

LiteVL: Efficient Video-Language Learning with Enhanced Spatial-Temporal Modeling

Dongsheng Chen¹, Chaofan Tao², Lu Hou³, Lifeng Shang³, Xin Jiang³, Qun Liu³

¹Peking University ²The University of Hong Kong ³Huawei Noah's Ark Lab

chendsh@stu.pku.edu.cn, cftao@connect.hku.hk

{houlu3, shang.lifeng, jiang.xin, qun.liu}@huawei.com

Abstract

Recent large-scale video-language pre-trained models have shown appealing performance on various downstream tasks. However, the pre-training process is computationally expensive due to the requirement of millions of video-text pairs and the redundant data structure of each video. To mitigate these problems, we propose LiteVL, which adapts a pre-trained image-language model BLIP into a video-text model directly on downstream tasks, without heavy pre-training. To enhance the temporal modeling lacking in the image-language model, we propose to add temporal attention modules in the image encoder of BLIP with dynamic temporal scaling. Besides the model-wise adaptation, we also propose a non-parametric pooling mechanism to adaptively reweight the fine-grained video embedding conditioned on the text. Experimental results on text-video retrieval and video question answering show that the proposed LiteVL even outperforms previous video-language pre-trained models by a clear margin, though without any video-language pre-training.

1 Introduction

The increasing popularity of videos in various social media has aroused the interest in efficient modeling of videos and their connections with other modalities like texts. Video-language modeling targets at learning a shared multimodal semantic space for videos and texts to facilitate downstream tasks like text-video retrieval and video question answering (VideoQA). Previous video-language modeling usually relies on pre-training on a large-scale video-text pair, via video-text contrastive learning (Luo et al., 2021; Li et al., 2022a; Gorti et al., 2022), video-text matching (Fan et al., 2019; Luo et al., 2020; Li et al., 2022a), masked language modeling (Devlin et al., 2019) and masked frame modeling (Sun et al., 2019); or extra object detectors to extract fine-grained visual features (Zhu and

Yang, 2020; Chen et al., 2020). However, both pre-training on video-text pairs and using an off-the-shelf detector are computationally expensive and inefficient. Inaccurate detection results on limited categories may also lead to inferior performance.

In the unimodal video domain, TimeSformer (Bertasius et al., 2021) fails to pre-train video encoder directly on large-scale video datasets, but obtains good performance by initializing from a pre-trained Transformer-based ViT (Dosovitskiy et al., 2021) image encoder and training additional temporal attention modules directly on downstream video tasks. Similarly, ViViT (Arnab et al., 2021) also takes advantage of the already well-learned spatial visual representation in a pre-trained ViT model, and effectively adapts it for video tasks by directly fine-tuning on comparatively small downstream video datasets.

Inspired by TimeSformer and ViViT, in this paper, we also consider extending an image-language pre-trained model for video-text tasks without pre-training on video-text pairs. This requires us to not only leverage the already-learned alignment between spatial visual information and text in the image-language models, but also capture the additional temporal dependency efficiently. Thus we propose a simple yet efficient video-language model LiteVL, initialized from a recent pre-trained image-language model BLIP, but with both model-wise and feature-wise enhancement of temporal information. For model-wise enhancement, we propose to explicitly insert temporal attention layers with learnable scalings into the original image backbone, which can be adjusted for each downstream task. For feature-wise enhancement, we design a non-parametric pooling method to learn fine-grained spatial-temporal video features conditioned on the text description.

Empirical results on various tasks demonstrate that the proposed model, LiteVL, outperforms previous state-of-the-art methods by a clear margin,

even without any video-language pre-training or the usage of object detectors. In particular, LiteVL achieves 50.8% R1 score on **MSRVTT-9k** dataset in the task of text-video retrieval, and 42.9% accuracy on **MSRVTT-QA** dataset in the task of video question answering. Visualizations also demonstrate that our LiteVL captures important spatial-temporal information with fine-grained video-text alignment.

2 Related Work

2.1 Vision Transformers

Transformers (Vaswani et al., 2017), originally designed for natural language tasks, have recently been applied to the computer vision domain to model images and videos (Dosovitskiy et al., 2021; Liu et al., 2021; Touvron et al., 2021; Wang et al., 2021; Han et al., 2021). ViT (Dosovitskiy et al., 2021) is one of the most representative vision transformers, which processes each image as a sequence of image patches, and achieves remarkable performances on various image tasks.

Compared with image tasks, video understanding is more challenging because the additional temporal dimension brings a more complicated spatial-temporal information. To model the intertwined dependency of the spatial and temporal dimensions efficiently, video Transformers TimeSformer (Bertasius et al., 2021) and ViViT (Arnab et al., 2021) use the parameters of a well-trained image Transformer for initialization, and further design different variants of spatial-temporal attention mechanisms to capture the spatial-temporal dependencies.

2.2 Video-Language Modeling

The core of video-language models lies in modeling the interaction between the two modalities. Depending on whether using video-text pairs for pre-training, existing video-language models can be divided into two categories.

The main branch of works explicitly designs the spatial-temporal structure in video encoders, and pre-train with abundant video-text pairs to directly align videos and texts. Among these works, ALPRO (Li et al., 2022a), Frozen (Bain et al., 2021), and BridgeFormer (Ge et al., 2022) use WebVid2M (Bain et al., 2021) which contains 2.5M video-text pairs collected from the web for pre-training. Image-text pairs like CC3M (Sharma et al., 2018) and VG (Krishna et al., 2016) are also often used to enhance the spatial information

in this alignment. NoiseEST (Amrani et al., 2021), VideoCLIP (Xu et al., 2021), and CLIP4Clip (Luo et al., 2021) pretrain the model with a large-scale dataset HowTo100M (Fan et al., 2019) which contains 136M video-text pairs. The other branch of works does not rely on video-text pre-training. Instead, they extend a pre-trained image-text model to extract video features, and directly fine-tune on downstream tasks. In ClipBERT (Lei et al., 2021), BLIP (Li et al., 2022b), and X-Pool (Gorti et al., 2022), each video is viewed as a collection of images, whose representations obtained from an image encoder are then used to represent the video for later interaction with the text.

3 Method

In this section, we propose a method to efficiently extend an image-language pre-trained model to a video-language model, without pre-training on video-text pairs or the use of object detectors.

In Section 3.1, we first introduce the model architecture. Then we propose to enhance the temporal information in both model-wise and feature-wise manners. For model-wise enhancement, we propose to insert temporal attention layers with learnable scalings into the original image backbone (Section 3.2). For feature-wise enhancement, we design a non-parametric pooling method to learn fine-grained spatial-temporal video features conditioned on the text description (Section 3.3).

3.1 Model Architecture

As is shown in the Figure 1, our framework contains three parts: a video encoder, a text encoder, and a video-grounded text encoder. We initialize our framework based on the recently proposed image-language model BLIP (Li et al., 2022b) trained over massive image-text pairs.

Video Encoder. To enhance the temporal dependency of the video encoder, following TimeSformer (Bertasius et al., 2021), we insert additional temporal attention modules into the original BLIP image encoder, whose weights are initialized with the original spatial attention modules (Figure 1a). We use the Divided Space-Time Attention proposed in TimeSformer. We first compute the temporal attention by comparing each patch with the patches at the same spatial location in different frames, and then compute the spatial attention in each frame separately.

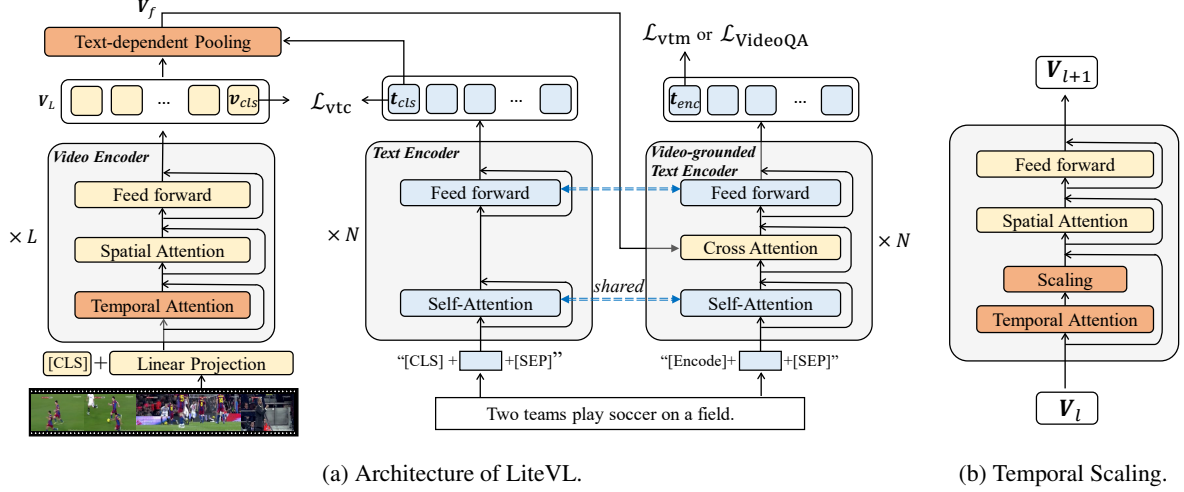


Figure 1: (a) The architecture of LiteVL. The model is initialized from the pre-trained image-language model BLIP, but is equipped with additional temporal attention modules and text-dependent pooling, to quickly adapt to video-language downstream tasks without pre-training. (b) The proposed dynamic temporal scaling, which adjusts the scale of the newly-added temporal attention according to each downstream task.

Text Encoder. The text encoder is a BERT model, initialized with the original text encoder in BLIP. A [CLS] token is appended to the beginning of the text sequence. This unimodal text encoder uses bi-directional self-attention and uses the output embedding of the [CLS] token to summarize the text sequence.

Video-grounded Text Encoder. This encoder shares the parameters with the unimodal text encoder. Moreover, to fuse the video features from the video encoder, one additional cross-attention layer is added between the self-attention layer and the feed-forward network for each transformer layer. Following BLIP (Li et al., 2022b), we also use a special [Encode] token before the text sequence, and its output embedding is used as the multimodal representation of the video-text pair.

3.2 Dynamic Temporal Scaling

We wish to preserve the spatial representation and its alignment with the text encoder learned by the image-language pre-trained model, as well as learn temporal expressivity for video-language tasks. As will be shown in Table 6, directly using TimeSformer yields better results than the original ViT. To provide more sufficient temporal expressiveness of the video encoder, we propose to learn a set of scalings that dynamically adjust the newly inserted temporal attention modules according to each specific task, as shown in Figure 1b.

Specifically, denote the output feature after the

temporal attention at the l -th Transformer layer as $\mathbf{V}_l^{\text{TAtn}} \in \mathbb{R}^{S \times T \times D}$. For t -th frame $[\mathbf{V}_l^{\text{TAtn}}]_t \in \mathbb{R}^{S \times D}$, we add a learnable scaling factor $\alpha_{l,t} \in \mathbb{R}$ with a tanh-gating mechanism as:

$$\alpha_{l,t} = \tanh(\gamma_{l,t}) + 1, \quad (1)$$

where $\gamma_{l,t}$ is a learnable scalar initialized at 0. Then the scaled output of temporal attention $\mathbf{V}_l^{\text{TAtn}}$ is calculated as:

$$[\mathbf{V}_l^{\text{TAtn}}]_t = \alpha_{l,t} \cdot [\mathbf{V}_l^{\text{TAtn}}]_t, \quad (2)$$

before the residual connection. Note the [CLS] token is kept but not involved in the computation of scaling. The choice of tanh-gating ensures that $\alpha_{l,t}$ ranges from 0 to 2. Initially, our model is equivalent to TimeSformer (i.e., $\alpha_{l,t}=1$, treat each frame equally), and then explicitly reweight the frames in each transformer block during the fine-tuning stage. When $\alpha_{l,t}$ reduces to 0, the video encoder degenerates to the ViT used in the original BLIP model, which does not consider any temporal dependency in extracting the video features.

3.3 Text-dependent Pooling

Before interacting with the textual modality via cross-attention or self-attention, previous methods directly concatenate the features from all frames with equal importance (Li et al., 2022b; Ge et al., 2022), or aggregate the video features with heuristic mean/max pooling methods spatially or temporally (Luo et al., 2021; Li et al., 2022a). However,

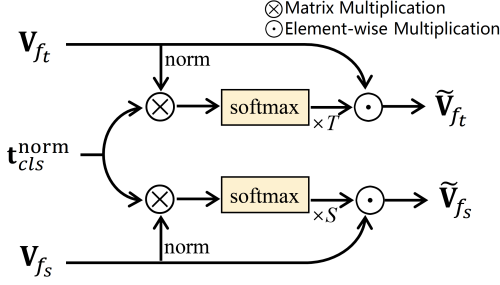


Figure 2: Illustration of text-dependent pooling. We reweight the pooled spatial and temporal video features based on the similarities between the normalized text feature $\mathbf{t}_{cls}^{\text{norm}}$ and visual features \mathbf{V}_{f_t} and \mathbf{V}_{f_s} .

not all frames or spatial positions are equally representative for the whole video, and different frames or positions have different semantic similarities to the textual query (e.g., textual description in text-video retrieval tasks or textual question in video question answering tasks). For example, given a video description “a golf player is trying to hit the ball into the pit”, the video encoder is expected to focus on the object of interest (i.e., ball) and the motion of hitting across the frames.

As illustrated in Figure 2, we design a non-parametric text-dependent pooling to reweight the video features spatially and temporally depending on the corresponding textual query, enabling fine-grained video-text alignment.

Specifically, given a video with T frames, each frame is patchified into S patches, and a [CLS] token is inserted before the ST patches. Denote the original output embedding of the video encoder as $\mathbf{V}_L \in \mathbb{R}^{(1+ST) \times D}$. $\mathbf{V}_{f_t} \in \mathbb{R}^{T \times D}$ and $\mathbf{V}_{f_s} \in \mathbb{R}^{S \times D}$ are the video features pooled by averaging \mathbf{V}_L along the spatial and temporal dimension, respectively. Note that the feature of [CLS] token is not involved in averaging. Denote $\mathbf{t}_{cls} \in \mathbb{R}^D$ as the output embedding of the [CLS] token obtained from the text encoder.

Intuitively, the more similar a visual feature is to the text description, the more representative it is for understanding the content of the whole video. Thus we compute the similarity between the ℓ_2 normalized features of each frame in $\mathbf{V}_{f_t}^{\text{norm}}$ and the text feature $\mathbf{t}_{cls}^{\text{norm}}$, and reweight the features in \mathbf{V}_{f_t} as:

$$\begin{aligned} \mathbf{g}_t &= \text{softmax}(\mathbf{V}_{f_t}^{\text{norm}} \mathbf{t}_{cls}^{\text{norm}} / \tau), \\ \tilde{\mathbf{V}}_{f_t} &= T \cdot \mathbf{g}_t \odot \mathbf{V}_{f_t}, \end{aligned} \quad (3)$$

where \odot means element-wise multiplication, and τ is the temperature which controls the sharpness

of the weight distribution. We multiply the weights from the softmax function by the number of frames T , such that the sum of total weights keeps the same as direct concatenation. Similarly, we compute the similarity between the ℓ_2 normalized features of each spatial position in $\mathbf{V}_{f_s}^{\text{norm}}$ and the text feature $\mathbf{t}_{cls}^{\text{norm}}$, and reweight the features in \mathbf{V}_{f_s} as:

$$\tilde{\mathbf{V}}_{f_s} = S \cdot \text{softmax}(\mathbf{V}_{f_s}^{\text{norm}} \mathbf{t}_{cls}^{\text{norm}} / \tau) \odot \mathbf{V}_{f_s}. \quad (4)$$

The final aggregated video feature to be fed to the video-grounded text encoder is a concatenation of $\tilde{\mathbf{V}}_{f_t}$, $\tilde{\mathbf{V}}_{f_s}$, and the original video feature \mathbf{V}_L :

$$\mathbf{V}_f = [\tilde{\mathbf{V}}_{f_t}, \tilde{\mathbf{V}}_{f_s}, \mathbf{V}_L]. \quad (5)$$

Remark 1. Besides using the text to reweight the aggregated features after spatial pooling (i.e., \mathbf{V}_{f_t}) and temporal pooling (i.e., \mathbf{V}_{f_s}), one simple baseline is to directly concatenate them with the original features \mathbf{V}_L to compose \mathbf{V}_f as: $\mathbf{V}_f = [\mathbf{V}_{f_t}, \mathbf{V}_{f_s}, \mathbf{V}_L] \in \mathbb{R}^{(1+T+S+ST) \times D}$. We dub it as **vanilla pooling**. Despite its simplicity, this pooling achieves competitive performance (Table 6).

3.4 Training Objectives

After obtaining the aggregated video features \mathbf{V}_f , we feed them to each cross-attention layer of the video-grounded text encoder. Consider a training batch with B video-text pairs. For the k -th video-text pair, denote the ℓ_2 normalized output embeddings of the [CLS] tokens from the video encoder and the text encoder as \mathbf{v}_{cls}^k and \mathbf{t}_{cls}^k ¹, respectively. The output embedding of the [Encode] token of the video-grounded text encoder is denoted as \mathbf{t}_{enc}^k .

Text-Video Retrieval. Contrastive learning alone has recently been found to learn better representations than its predictive counterpart in multi-modal pre-training (Radford et al., 2021). When used together with the predictive counterpart (Li et al., 2022a), it also boosts the performance. To align the video encoder and text encoder, we also utilize both the contrastive and predictive learning objectives. We apply contrastive learning over the output representations of the video encoder and the text encoder by

¹Following BLIP (Li et al., 2022b), before normalization, the [CLS] tokens from the video and text encoders are mapped to a lower-dimensional space through separate linear projections. With a slight abuse of notation, this \mathbf{t}_{cls}^k has reduced dimension and is not the same as $\mathbf{t}_{cls}^{\text{norm}}$ in Eq.(3).

Methods	Pre-Training Data	MSRVTT-7k				MSRVTT-9k				DiDeMo			
		R1↑	R5↑	R10↑	MdR↓	R1↑	R5↑	R10↑	MdR↓	R1↑	R5↑	R10↑	MdR↓
BLIP [†]	14M	33.6	55.5	64.8	4	33.6	55.5	64.8	4	35.6	58.4	66.4	3
BLIP ^{384†}	14M	30.8	52.2	62.2	5	30.8	52.2	62.2	5	31.6	58.2	65.8	3
BLIP	14M	42.5	68.2	79.3	2	44.1	70.9	80.1	2	51.3	78.1	84.6	1
LiteVL _S	-	44.5	70.3	80.2	2	46.7	71.8	81.7	2	53.7	79.6	87.0	1
BLIP [†]	129M	41.4	63.3	72.8	2	41.4	63.3	72.8	2	40.0	63.1	72.4	2
BLIP ^{384†}	129M	40.5	62.8	71.4	2	40.5	62.8	71.4	2	37.3	61.7	69.4	3
BLIP [†] _{coco}	129M	40.3	63.6	72.1	2	40.3	63.6	72.1	2	40.0	64.6	73.1	2
BLIP ^{384†} _{coco}	129M	43.3	65.6	74.7	2	43.3	65.6	74.7	2	43.2	69.3	75.9	2
BLIP	129M	48.2	74.3	82.9	2	49.7	74.7	83.8	2	51.6	79.8	86.8	1
LiteVL _L	-	48.9	74.5	83.6	2	50.8	76.3	84.4	1	53.4	80.7	87.0	1
LiteVL ³⁸⁴ _{Lcoco}	-	49.7	75.5	83.3	2	51.5	75.9	85.7	1	53.2	79.6	87.5	1

Table 1: Comparison of LiteVL and BLIP on text-video retrieval tasks. The default resolution is 224×224 per video frame, and the superscript “384” means increasing it to 384×384 . The subscript “coco” means training with an extra COCO retrieval dataset. [†] means zero-shot inference used in BLIP by default.

optimizing a symmetric InfoNCE loss. The video-to-text contrastive loss \mathcal{L}_{v2t} is:

$$\mathcal{L}_{v2t} = -\frac{1}{B} \sum_{k=1}^B \log \frac{\exp(\mathbf{v}_{cls}^{k\top} \mathbf{t}_{cls}^k / \tau_c)}{\sum_j \exp(\mathbf{v}_{cls}^{k\top} \mathbf{t}_{cls}^j / \tau_c)},$$

where τ_c is a learnable temperature parameter initialized as 0.07. Similarly, the text-to-video contrastive loss \mathcal{L}_{t2v} is:

$$\mathcal{L}_{t2v} = -\frac{1}{B} \sum_{k=1}^B \log \frac{\exp(\mathbf{v}_{cls}^{k\top} \mathbf{t}_{cls}^k / \tau_c)}{\sum_j \exp(\mathbf{v}_{cls}^{j\top} \mathbf{t}_{cls}^k / \tau_c)}.$$

The video-text contrastive loss is defined as:

$$\mathcal{L}_{vtc} = \frac{1}{2} (\mathcal{L}_{v2t} + \mathcal{L}_{t2v}). \quad (6)$$

Following Li et al. (2022a), besides the contrastive loss, we also use a video-text matching loss \mathcal{L}_{vtm} , which predicts whether a pair of video and text is matched or not. For the k -th video-text pair, we map the joint video-text embedding \mathbf{t}_{enc}^k to a two-class probability \mathbf{p}_{vtm}^k , and calculate \mathcal{L}_{vtm} as:

$$\mathcal{L}_{vtm} = \frac{1}{B} \sum_{k=1}^B \text{CE}(\mathbf{y}_{vtm}^k, \mathbf{p}_{vtm}^k), \quad (7)$$

where \mathbf{y}_{vtm}^k is a 2-dimensional one-hot vector representing the ground-truth label, and $\text{CE}(\cdot, \cdot)$ is cross-entropy loss. The in-batch negatives used for \mathcal{L}_{vtm} are mined based on the contrastive similarity following Li et al. (2021). The overall training objective is:

$$\mathcal{L}_{\text{retrieval}} = \mathcal{L}_{vtc} + \mathcal{L}_{vtm}. \quad (8)$$

Video Question Answering. Following Fan et al. (2019), we view video question answering as a multimodal classification task and add a two-layer MLP classification head on the multimodal [Encode] token. Assume the number of answers is K , for the k -th video-text pair, we map the joint video-text embedding \mathbf{t}_{enc}^k to a K -class probability \mathbf{p}_{ans}^k , the training objective for video question answering is:

$$\mathcal{L}_{\text{VideoQA}} = \frac{1}{B} \sum_{k=1}^B \text{CE}(\mathbf{y}_{ans}^k, \mathbf{p}_{ans}^k), \quad (9)$$

where \mathbf{y}_{ans}^k is a K -dimensional one-hot classification label.

4 Experiments

In this section, we evaluate the efficacy of the proposed LiteVL on the text-video retrieval and video question answering (VideoQA) tasks. We initialize the weights of LiteVL from BLIP (Li et al., 2022b), which uses a ViT-B/16 as the image encoder, a BERT_{base} as the text encoder with additional cross-attention layers for the image-grounded text encoder. We use both BLIP variants pre-trained on 14M and 129M image-text pairs, respectively, and the corresponding LiteVL initialized from them are dubbed as LiteVL_S and LiteVL_L, respectively.

During fine-tuning, we randomly sample 8 and 16 frames per video for retrieval and VideoQA tasks, respectively. While in the inference stage, the frames are uniformly sampled. Following previous works (Ge et al., 2022; Gorti et al., 2022) and BLIP’s pre-training setting, we resize each of the

raw frames to 224×224 before feeding them into the model. For the text-dependent pooling, the temperature τ in Eq.(3) is set to 1.0 by default. More detailed training details and hyperparameters are in Appendix A.

4.1 Text-Video Retrieval

Datasets and Metrics. We finetune on two text-video retrieval datasets: (i) **MSRVTT** (Xu et al., 2016) consists of 10k videos and 200k text captions. Each video is paired with about 20 manually-labeled captions, and lasts about 10 to 32 seconds. There are two widely used ways to split the dataset, i.e., **MSRVTT-7k** (Miech et al., 2019) and **MSRVTT-9k** (Gabeur et al., 2020), which have 7k videos and 9k videos for training, respectively. For a comprehensive comparison with previous works, we use both splits that share the same 1k testing videos (Bain et al., 2021). (ii) **DiDeMo** (Hendricks et al., 2017) consists of 10k Flickr videos annotated with 40k text captions. We evaluate text-video retrieval following Lei et al. (2021), where all captions for the same video are concatenated into a single query.

We evaluate text-video retrieval by R@k and MdR following Bain et al. (2021). R@k means the recall (%) through k retrieval efforts. MdR represents the median rank of the retrieved video.

Comparison with BLIP. Previous BLIP concatenates the image features of all frames as the aggregated video feature and feeds it to the image-grounded text encoder. In Table 1, we show the comparison between our proposed LiteVL and the original BLIP as well as its variants with increased resolution (i.e., 384×384), pre-training on COCO (Lin et al., 2014) and fine-tuning setting. As can be seen, though inherited from the BLIP model, our proposed LiteVL clearly outperforms the original BLIP due to the explicit temporal modeling in both model-wise and feature-wise manners. In particular, LiteVL_S improves the R1 of the best-performed BLIP (14M) variant by 2.0, 2.6, and 2.4 points on **MSRVTT-7k**, **MSRVTT-9k**, and **DiDeMo**, respectively.

Comparison with Other Methods. Table 2, Table 3 and Table 4 show the comparison between LiteVL and recent methods on text-video retrieval on **MSRVTT-7k**, **MSRVTT-9k** and **DiDeMo**, respectively. On all three datasets, LiteVL surpasses previous works by a clear margin, including methods requiring heavy video-text pre-training (e.g.,

Methods	R1↑	R5↑	R10↑	MdR↓
NoiseEST (Amrani et al., 2021)	17.4	41.6	53.6	8
ClipBERT (Lei et al., 2021)	22.0	46.8	59.9	6
VideoClip (Xu et al., 2021)	30.9	55.4	66.8	-
ALPRO (Li et al., 2022a)	33.9	60.7	73.2	3
CLIP4Clip (Luo et al., 2021)	42.1	71.9	81.4	2
BLIP [†] (Li et al., 2022b)	43.3	65.6	74.7	2
X-Pool (Gorti et al., 2022)	43.9	72.5	82.3	2
LiteVL _S	44.5	70.3	80.2	2
LiteVL _L	48.9	74.5	83.6	2

Table 2: Results of text-video retrieval on the test split of **MSRVTT-7k**. † means zero-shot results reported by the original BLIP paper.

Methods	R1↑	R5↑	R10↑	MdR↓
CE (Liu et al., 2019)	20.9	48.8	62.4	6
Frozen (Bain et al., 2021)	31.0	59.5	70.5	3
BridgeFormer (Ge et al., 2022)	37.6	64.8	75.1	3
CLIP4Clip (Luo et al., 2021)	44.5	71.4	81.6	2
BLIP [†] (Li et al., 2022b)	43.3	65.6	74.7	2
X-Pool (Gorti et al., 2022)	46.9	72.8	82.2	2
LiteVL _S	46.7	71.8	81.7	2
LiteVL _L	50.8	76.3	84.4	1

Table 3: Results of text-video retrieval on the test split of **MSRVTT-9k** dataset.

Methods	R1↑	R5↑	R10↑	MdR↓
CE (Liu et al., 2019)	16.1	41.1	82.7	8
ClipBERT (Lei et al., 2021)	20.4	48.0	60.8	6
Frozen (Bain et al., 2021)	34.6	65.0	74.7	3
ALPRO (Li et al., 2022a)	35.9	67.5	78.8	3
BridgeFormer (Ge et al., 2022)	37.0	62.2	73.9	3
CLIP4Clip (Luo et al., 2021)	43.4	70.2	80.6	2
LiteVL _S	53.7	79.6	87.0	1
LiteVL _L	53.4	80.7	87.0	1

Table 4: Results of text-video retrieval on the test split of **DiDeMo** dataset.

ALPRO, CLIP4Clip and BridgeFormer) or based on image-language pre-trained models (e.g., ClipBERT and X-Pool). Note that X-Pool also uses a parametric text-dependent pooling method to aggregate video features from an image-language pre-trained model. However, our proposed LiteVL still outperforms it with the simpler non-parametric similarity-based text-dependent pooling and the proposed dynamic scaling. Besides, LiteVL_L performs significantly better than LiteVL_S on both splits of MSRVTT, and similarly on DiDeMo. This indicates that the general multimodal priors learned from a large-scale image-text dataset can also benefit text-video retrieval tasks.

4.2 Video Question Answering

Datasets and Metrics. (i) **MSRVTT-QA** (Xu et al., 2017) consists of 10k videos, 243k open-ended questions, and 1.5k answer candidates based on MSRVTT. (ii) **MSVD-QA** (Xu et al., 2017) is based on MSVD (Chen and Dolan, 2011), and includes 1,970 videos, 50k question-answer pairs, and 2,423 answer candidates. We use top-1 accuracy (%) as the evaluation metric. For both datasets, we use the same train/val/test splits as Xu et al. (2017), and select the model with the highest accuracy on the validation set for testing.

Methods	MSRVTT-QA	MSVD-QA
HME (Fan et al., 2019)	33.0	33.7
HGA (Jiang and Han, 2020)	35.5	34.7
ClipBERT (Lei et al., 2021)	37.4	-
SSML (Amrani et al., 2021)	35.1	35.1
CoMVT (Seo et al., 2021)	39.5	42.6
VQA-T (Yang et al., 2021)	41.5	46.3
ALPRO (Li et al., 2022a)	42.1	45.9
LiteVL _S (Ours)	42.9	47.3
LiteVL _L (Ours)	42.5	42.9

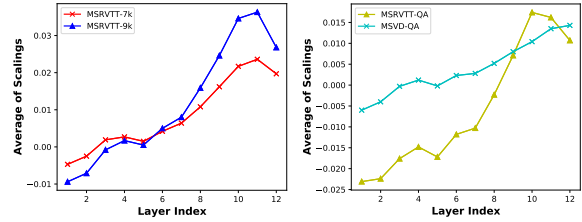
Table 5: Top-1 accuracy (%) of VideoQA on the test set of **MSRVTT-QA** and **MSVD-QA**.

Comparison with Other Methods. In Table 5, we compare our proposed LiteVL with existing methods on the open-ended video question answering datasets **MSRVTT-QA** and **MSVD-QA**. Again, LiteVL outperforms all the compared methods, including VQA-T (Yang et al., 2021) pre-trained with 69M VideoQA samples and the recent ALPRO pre-trained with 5.5M video-text pairs. Instead, LiteVL requires no general or QA-related video-text pre-training. Interestingly, LiteVL_L is initialized from BLIP pre-trained on more image-text pairs than LiteVL_S, but performs worse on two VideoQA tasks. We speculate this is because the additional 115M image-text pairs used to train BLIP (129M) crawled from the web or filtered by BLIP captioner are noisy and may bias towards the captioner. This may mislead the VideoQA task which requires more precise multimodal reasoning to answer questions than retrieval task.

4.3 Ablation Study

4.3.1 Dynamic Temporal Scaling

In Table 6, we compare the proposed dynamic temporal scaling against using (i) constant scaling $\alpha_{l,t} = 1$ in Eq.(2), which reduces to directly using TimeSformer as the video encoder; and (ii)



(a) Text-video retrieval.

(b) VideoQA.

Figure 3: Average temporal scalings $\gamma_{l,t}$ for different frames of each layer in the video encoder of LiteVL_S trained on different tasks.

constant scaling $\alpha_{l,t} = 0$ in Eq.(2), which reduces to using the ViT image encoder as the video encoder. As can be seen, the proposed dynamic scaling learned upon each task performs better than the two special cases. By adopting TimeSformer ($\alpha_{l,t} = 1$) instead of ViT ($\alpha_{l,t} = 0$), the performance is boosted since the temporal dependency is considered via the additional temporal attention module. With the proposed lightweight temporal scaling to adjust the frame-level importance according to each specific task, the performance is further improved.

Visualization. As shown in Figure 3, we visualize the average of the learned scalings $\gamma_{l,t}$ of each layer in the video encoder for both retrieval (i.e., **MSRVTT-7k**, **MSRVTT-9k**) and VideoQA (i.e., **MSRVTT-QA**, **MSVD-QA**) tasks. For all datasets, the average scaling is lower than 0 at the first layer and then shows an upward trend as the depth increases. This indicates that the shallow layers focus more on understanding the content of each frame, and pay less attention to temporal dependency among different frames. When the depth increases, the spatial feature of each frame becomes more global (Dosovitskiy et al., 2021), and the model gradually seeks to learn the temporal dependencies among them.

4.3.2 Text-dependent Pooling

In Table 6, we compare our proposed text-dependent pooling in Section 3.3 against several baseline pooling methods using different combinations of the original features \mathbf{V}_f , spatially pooled features \mathbf{V}_{f_t} and temporally pooled features \mathbf{V}_{f_s} . As can be seen, compared with using only the original features, using either additional spatially or temporally pooled features improves the performance, and combining both of them further boosts performance. When coupled with the reweighting

Methods	MSRVTT-7k				DiDeMo				MSVD-QA
	R1↑	R5↑	R10↑	MdR↓	R1↑	R5↑	R10↑	MdR↓	Acc.↑
LiteVL _S	44.5	70.3	80.2	2	53.7	79.6	87.0	1	47.3
<i>Dynamic Temporal Scaling</i>									
w/o Scaling ($\alpha_{t,l} = 1$)	44.4	69.4	79.5	2	52.8	80.3	87.0	1	47.1
w/o Temporal attention ($\alpha_{t,l} = 0$)	42.8	68.6	79.4	2	51.7	78.5	85.6	1	45.5
<i>Text-dependent Pooling</i>									
Original features	42.8	68.8	79.8	2	51.4	77.9	85.7	1	45.3
Original features + Temporal pooling	43.0	69.5	79.8	2	52.2	79.1	86.5	1	46.3
Original features + Spatial pooling	43.0	69.1	79.5	2	51.7	78.9	85.9	1	45.9
Original features + Temporal pooling + Spatial pooling	43.6	69.8	79.8	2	52.3	79.3	86.5	1	46.4

Table 6: Ablation studies on dynamic temporal scaling and text-dependent pooling of the proposed LiteVL.

mechanism in Section 3.3, our proposed LiteVL obtains the best performance.

In addition, since the visually or temporally pooled features have much smaller sizes than the original features, using them merely increases the computation or memory cost of the cross-attention module of the video-grounded text encoder. The extra computation or memory cost incurred here is theoretically relatively acceptable.

Effect of Temperature in the Text-dependent Pooling. We vary the temperature τ between 0.01 and 5.0 of the text-dependent pooling to study the effect of the temperature in the text-dependent pooling. As is shown in Table 7, when τ equals 1.0, both text-video retrieval and video question answering achieve the best performance. Therefore, the temperature τ of this pooling method is set to 1.0 for all datasets by default.

Visualization. To better understand the effect of text-dependent pooling, we use LiteVL_S to visualize the video-text pair from MSRVTT-7k testing set and their corresponding temporal weights (g_t in Eq.(3)). As show in Figure 4, when the changes among different frames are relatively large, the proposed text-dependent pooling encourages the model to assign higher weights to frames better described by the caption. For instance, in the first example, the second and fourth frames are more related to the caption “Three kids sing together on the voice.” and assigned higher weights.

On the contrary, as can be seen from the last two examples in Figure 4, when the different frames only differ in minor changes and each frame is similarly close to the caption, the learned weights for each frame are also similar. For these cases, we further study the more fine-grained spatial-temporal dependencies using the Grad-CAM (Sel-

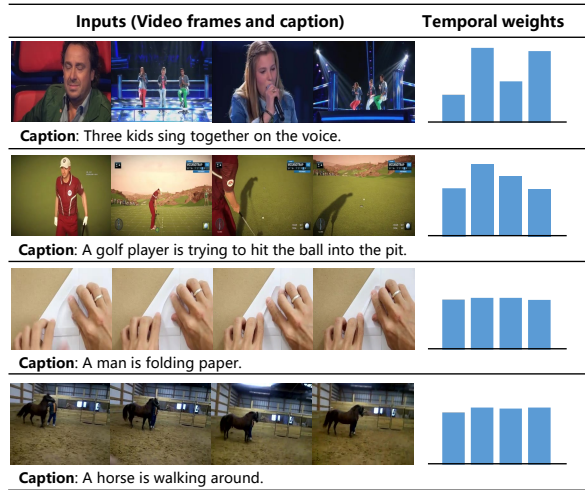


Figure 4: Bar plots of temporal weights in text-dependent pooling. Video frames more related to the caption are assigned with higher weights.

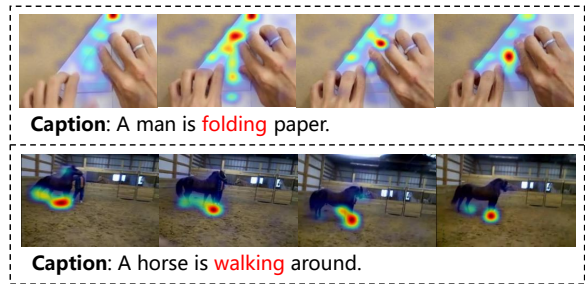


Figure 5: Grad-CAM visualizations on the cross-attention maps corresponding to highlighted keywords.

varaju et al., 2017) visualizations. We compute Grad-CAM using the cross-attention maps averaged over all attention heads in the 8-th layer (a specialized layer in grounding) of the video-grounded text encoder. The gradients are acquired by maximizing the video-text matching score in Eq.(7). As can be seen in Figure 5, the proposed LiteVL effectively captures the minor changes among dif-

Pooling Methods	τ	MSRVTT-7k				DiDeMo				MSVD-QA
		R1 \uparrow	R5 \uparrow	R10 \uparrow	MdR \downarrow	R1 \uparrow	R5 \uparrow	R10 \uparrow	MdR \downarrow	Acc. \uparrow
Text-dependent pooling	0.01	8.2	24.7	35.1	23	6.7	23.5	34.2	32	27.4
	0.1	44.3	69.7	79.0	2	52.1	80.0	86.9	1	46.1
	1.0	44.4	69.4	79.5	2	52.8	80.3	87.0	1	47.1
	2.0	43.9	69.1	79.0	2	52.3	79.8	86.1	1	46.3
	5.0	44.1	69.3	78.7	2	50.7	79.6	86.3	1	46.4

Table 7: Effect of different temperatures (τ) in the text-dependent pooling on LiteVL_S.

Methods	MSRVTT-7k				MSRVTT-9k				DiDeMo			
	R1 \uparrow	R5 \uparrow	R10 \uparrow	MdR \downarrow	R1 \uparrow	R5 \uparrow	R10 \uparrow	MdR \downarrow	R1 \uparrow	R5 \uparrow	R10 \uparrow	MdR \downarrow
LiteVL _S	44.5	70.3	80.2	2	46.7	71.8	81.7	2	53.7	79.6	87.0	1
w/ top- k	44.3	69.9	80.1	2	46.4	72.0	81.1	2	53.5	79.5	87.1	1
LiteVL _L	48.9	74.5	83.6	2	50.8	76.3	84.4	1	53.4	80.7	87.0	1
w/ top- k	48.4	75.4	82.5	2	50.5	76.5	84.1	1	52.3	79.2	86.8	1

Table 8: Effect of using s_{vtc} to filter top- k ($k=100$) candidates and calculate their s_{vtm} score for ranking.

ferent frames. This also indicates that our proposed text-dependent pooling provides fruitful information for the video-grounded text encoder. More visualizations are in Appendix C.

5 Discussions

5.1 Extension to Other Image-language Pre-trained Models

In this work, we choose BLIP to initialize our proposed model mainly because (i) it performs well on various downstream image-language tasks; and (ii) it can be regarded as a single-stream and dual-stream hybrid structure. Its dual-stream part allows efficient inference for cross-modal retrieval tasks, while its cross-attention allows deep cross-modal interaction for tasks like VQA.

On the other hand, the proposed dynamic temporal scaling and text-dependent pooling can also be applied to the dual-stream model like CLIP (Radford et al., 2021). For this setting, we also conduct a simple experiment. For CLIP, we use the proposed text-dependent pooling on top of the video features. As CLIP relies on the global features for retrieval, instead of concatenation in Eq.(5), we compute a weighted average of the reweighted features. Compared with a recent work CLIP4Clip which also extends CLIP for video retrieval, CLIP with our proposed method improves the best CLIP4Clip-meanP method by 1.9% and 1.7%, for the R1 and R10 on MSRVTT-7k, respectively.

5.2 Scaling to Larger-scale Retrieval Tasks

Since the test set sizes for all three retrieval datasets used in this work are relatively small, we compute a pairwise VTM score s_{vtm} for all video-text pairs during inference. However, the speed of inference in this approach will be slow when the size of dataset is huge in real-world scenarios.

In this section, we provide a more efficient retrieval solution. Specifically, we first compute the video-text similarity score s_{vtc} for all video-text pairs, then we take the top- k candidates and calculate their VTM score for ranking. Such method can speed up inference, because the k can be set to be very small compared with the test set size. Table 8 shows that using this efficient two-stage retrieval solution has negligible performance degradation.

6 Conclusion

We propose LiteVL, a video-language model without heavy video-language pre-training or object detectors. LiteVL inherits the already-learned alignment between the spatial visual information and textual information, from a pre-trained image-language model. Then, an extra temporal attention with dynamic temporal scaling is proposed to learn the temporal dynamics in the video. We also introduce a non-parametric pooling method which aggregates video features conditioned on the text description, enabling fine-grained video-language alignment. Empirical results show that our LiteVL outperforms the state-of-the-art methods trained with much more training data.

Acknowledgements

We gratefully acknowledge the support of [MindSpore²](#) and Ascend AI Processor for this research.

Limitations

Although LiteVL achieves great performance without the video-language pre-training, it is interesting to explore how to combine a few video-language pairs from the pre-training datasets to further boost the performance. In addition, we view VideoQA as a classification task, which may be limited to a fixed number of answer candidates. In the future, we would like to generation ability of the proposed LiteVL on more diverse long-form visual question answering datasets.

Ethics Statement

During the training process, the knowledge from the inherited image-text pre-trained models and the training data may have unsuitable information. Therefore, our trained models are possibly exposed to the similar risks of large language models as mentioned in [Weidinger et al. \(2021\)](#). Thanks to [Thoppilan et al. \(2022\)](#), harmful training data can be filtered to improve the safety of our model. On the other hand, there is no generative task involved in our video-text understanding framework, which means our framework will not output offensive language.

Before deploying the models in real-world applications, risk analysis is necessary since the used data may contain some unintended privacy information and improper video/text that do not conform to social norms.

References

Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex M. Bronstein. 2021. [Noise estimation using density estimation for self-supervised multimodal learning](#). In *AAAI*.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. 2021. [Vivit: A video vision transformer](#). In *ICCV*.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. [Frozen in time: A joint video and image encoder for end-to-end retrieval](#). In *ICCV*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. [Is space-time attention all you need for video understanding?](#) In *ICML*.

²<https://www.mindspore.cn/>

David L. Chen and William B. Dolan. 2011. [Collecting highly parallel data for paraphrase evaluation](#). In *ACL*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [Uniter: Universal image-text representation learning](#). In *ECCV*.

Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc Le. 2020. [Randaugment: Practical automated data augmentation with a reduced search space](#). In *NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *NAACL*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *ICLR*.

Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. 2019. [Heterogeneous memory enhanced multimodal attention model for video question answering](#). In *CVPR*.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. [Multi-modal transformer for video retrieval](#). In *ECCV*.

Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. 2022. [Bridgeformer: Bridging video-text retrieval with multiple choice questions](#). In *CVPR*.

Satya Krishna Gorti, Noel 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 *CVPR*.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. 2021. [Transformer in transformer](#). In *NeurIPS*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. 2017. [Localizing moments in video with natural language](#). In *ICCV*.

Pin Jiang and Yahong Han. 2020. [Reasoning with heterogeneous graph alignment for video question answering](#). In *AAAI*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2016. [Visual genome: Connecting language and vision using crowdsourced dense image annotations](#). *CoRR*, abs/1602.07332.

- 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 *CVPR*.
- Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven C. H. Hoi. 2022a. [Align and prompt: Video-and-language pre-training with entity prompts](#). In *CVPR*.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022b. [BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation](#). In *ICML*.
- Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021. [Align before fuse: Vision and language representation learning with momentum distillation](#). In *NeurIPS*.
- Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft COCO: common objects in context](#). In *ECCV*.
- Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. [Use what you have: Video retrieval using representations from collaborative experts](#). In *BMVC*.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. [Swin transformer: Hierarchical vision transformer using shifted windows](#). In *ICCV*.
- Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. [Univilm: A unified video and language pre-training model for multimodal understanding and generation](#). *CoRR*, abs/2002.06353.
- Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. [Clip4clip: An empirical study of CLIP for end to end video clip retrieval](#). *CoRR*, abs/2104.08860.
- Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. [Howto100m: Learning a text-video embedding by watching hundred million narrated video clips](#). In *ICCV*.
- MindSpore. <https://www.mindspore.cn>. 2021.
- 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. 2021. [Learning transferable visual models from natural language supervision](#). In *ICML*.
- Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. [Grad-cam: Visual explanations from deep networks via gradient-based localization](#). In *ICCV*.
- Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. 2021. [Look before you speak: Visually contextualized utterances](#). In *CVPR*.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. [Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning](#). In *ACL*.
- Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. [Videobert: A joint model for video and language representation learning](#). In *ICCV*.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. [Lamda: Language models for dialog applications](#). *CoRR*, abs/2201.08239.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. [Training data-efficient image transformers & distillation through attention](#). In *ICML*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *NeurIPS*.
- Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. [Pyramid vision transformer: A versatile backbone for dense prediction without convolutions](#). In *ICCV*.
- Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. [Ethical and social risks of harm from language models](#). *CoRR*, abs/2112.04359.
- Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. [Video question answering via gradually refined attention over appearance and motion](#). In *ACM MM*.
- Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. [Video-clip: Contrastive pre-training for zero-shot video-text understanding](#). In *EMNLP*.
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. [MSR-VTT: A large video description dataset for bridging video and language](#). In *CVPR*.
- Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2021. [Just ask: Learning to answer questions from millions of narrated videos](#). In *ICCV*.
- Linchao Zhu and Yi Yang. 2020. [Actbert: Learning global-local video-text representations](#). In *CVPR*.

A Detailed Fine-tuning Setups

We list the detailed fine-tuning setups on each dataset in Table 9 and Table 10. For all downstream datasets, we resize each frame to 224×224 unless otherwise stated. Following ALPRO (Li et al., 2022a), we randomly select N_v frames from the video, with parameters $\frac{N_v}{2}$ frames from the first and second half of the video, respectively. We use RandomAugment (Cubuk et al., 2020) on the frames sampled from each video. For all experiments, we use the same random seed (e.g., 42) to ensure reproduction.

Config	MSRVTT	DiDeMo
Optimizer	AdamW	AdamW
Init learning rate (lr)	2.5e-5	4e-5
Scaling learning rate	$1.25 \times lr$	$1.25 \times lr$
Weight decay	1e-3	1e-3
Optimizer momentum	$\beta_1, \beta_2=0.9, 0.98$	$\beta_1, \beta_2=0.9, 0.98$
Lr schedule	linear decay	linear decay
Warmup ratio	0.1	0.1
Batch size	64	96
Init τ_c	0.07	0.07
Text length	40	50
Frame number	8	8
Training epochs	5	10
Augmentation	RandAug(2,5)	RandAug(2,5)

Table 9: Experimental setup for **MSRVTT** (both 7k and 9k splits) and **DiDeMo** text-video retrieval.

B Comparison with Previous Work

To align video and text features, previous approaches can be generally divided into two categories. On the one hand, common dual-stream models CLIP4Clip (Luo et al., 2021) fuse *global* video feature from global mean pooling or the [CLS] token, and then interact fused video feature with text feature based on a simple multilayer perceptron head on the top. On the other hand, the cross-attention module is adopted where the key/value are obtained from the aggregated video feature, and the query is obtained from the text feature. Previous methods mainly use two ways to aggregate the original output video features V_L into V_f in the video-grounded text encoder: (i) keep the *original* features without modification; (ii) apply mean pooling over the *spatial* or *temporal* dimension.

We provide a more detailed comparison with related works in Table 11. We list how previous works extract the video features used for the alignment with text features. In addition, the video

Config	MSRVTT-QA	MSVD-QA
Optimizer	AdamW	AdamW
Init learning rate (lr)	5e-5	5e-5
Scaling learning rate	$1.25 \times lr$	$1.0 \times lr$
Weight decay	1e-3	1e-3
Optimizer momentum	$\beta_1, \beta_2=0.9, 0.98$	$\beta_1, \beta_2=0.9, 0.98$
Lr schedule	linear decay	linear decay
Warmup ratio	0.1	0.1
Batch size	96	96
Init τ_c	0.07	0.07
Text length	40	40
Frame number	16	16
Training epochs	10	15
Augmentation	RandAug(2,5)	RandAug(2,5)

Table 10: Experimental setup for fine-tuning on **MSRVTT-QA** and **MSVD-QA**.

encoder, text encoder, and pre-training data used by different methods are also provided.

C More Qualitative Results

We provide more visualizations of temporal weights g_t in Figure 6. To better understand how text-dependent pooling affects the decision, we take a closer look at when the proposed text-dependent pooling changes the decision over vanilla pooling (Remark 1). We find that the temporal weights of the changed decisions have a clearly higher standard deviation than the unchanged ones, indicating that text-dependent pooling tends to change the decisions when the different frames are more dissimilar. For instance, for the first case in Figure 6, its caption “The girl shows the boys her medal in this cartoon” is mainly related to the middle two frames. By assigning higher importance to these two frames, the proposed text-dependent pooling makes a correct decision while the vanilla pooling fails.

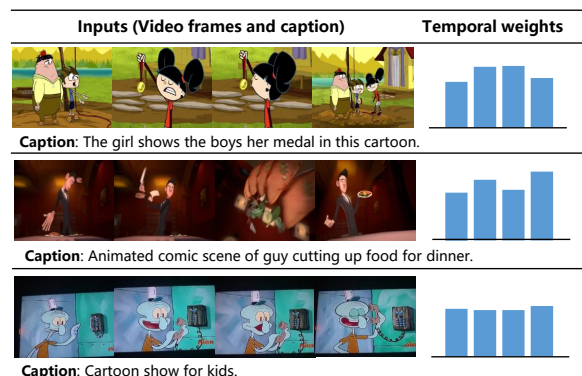


Figure 6: Bar plot of temporal weights learned by the text-dependent pooling.

Methods	Pooling Method	Video Encoder	Text Encoder	Pre-training Data
HT100M Miech et al. (2019)	Temporal	2D/3D CNN	word2vec	HowTo100M
NoiseEST. (Amrani et al., 2021)	Temporal	2D/3D CNN	word2vec	Howto100M
ClipBERT (Lei et al., 2021)	Temporal	2D CNN	BERT	COCO, VG
VideoClip (Xu et al., 2021)	Global (Average)	3D CNN (S3D)	BERT	HowTo100M
ALPRO (Li et al., 2022a)	Temporal	TimeSformer	BERT	CC3M, WebVid2M
CLIP4Clip (Luo et al., 2021)	Temporal	ViT (CLIP)	BERT (CLIP)	HowTo100M
BLIP (Li et al., 2022b)	Original	ViT	BERT	COCO, VG, CC3M, CC12M, SBU
Frozen (Bain et al., 2021)	Global ([CLS])	TimeSformer	DistilBERT	CC3M, WebVid2M
BridgeFormer (Ge et al., 2022)	Original	ViT	DistilBERT	CC3M, WebVid2M
VQA-T (Yang et al., 2021)	Global ([CLS])	3D CNN	DistilBERT	HowToVQA69M
LiteVL (Ours)	Spatial + Temporal + Original	TimeSformer	BERT	-

Table 11: Comparison between previous works with ours in terms of pooling method, video encoder, text encoder and pre-training dataset. The pooling method refers to the way to aggregate the video feature based on the original video feature $\mathbf{V}_L \in \mathbb{R}^{(1+ST) \times D}$ (Original), before the alignment with text feature. “Spatial” and “Temporal” denote the spatially-pooled feature $\mathbf{V}_{f_t} \in \mathbb{R}^{T \times D}$, and temporally-pooled feature $\mathbf{V}_{f_s} \in \mathbb{R}^{S \times D}$ from \mathbf{V}_L .