Revisiting Temporal Modeling for CLIP-based Image-to-Video Knowledge Transferring

Ruyang Liu[*1,3◇] Jingjia Huang[*2] Ge Li[1] Jiashi Feng[2] Xinglong Wu[2] Thomas H. Li ✉[1]
[1]School of Electronic and Computer Engineering, Peking University [2]ByteDance Inc [3]Peng Cheng Laboratory
{ruyang@stu,geli@ece,thomas@}.pku.edu.cn {huangjingjia,jshfeng,wuxinglong}@bytedance.com

Abstract

Image-text pretrained models, e.g., CLIP, have shown impressive general multi-modal knowledge learned from large-scale image-text data pairs, thus attracting increasing attention for their potential to improve visual representation learning in the video domain. In this paper, based on the CLIP model, we revisit temporal modeling in the context of image-to-video knowledge transferring, which is the key point for extending image-text pretrained models to the video domain. We find that current temporal modeling mechanisms are tailored to either high-level semantic-dominant tasks (e.g., retrieval) or low-level visual pattern-dominant tasks (e.g., recognition), and fail to work on the two cases simultaneously. The key difficulty lies in modeling temporal dependency while taking advantage of both high-level and low-level knowledge in CLIP model. To tackle this problem, we present Spatial-Temporal Auxiliary Network (STAN) – a simple and effective temporal modeling mechanism extending CLIP model to diverse video tasks. Specifically, to realize both low-level and high-level knowledge transferring, STAN adopts a branch structure with decomposed spatial-temporal modules that enable multi-level CLIP features to be spatial-temporally contextualized. We evaluate our method on two representative video tasks: Video-Text Retrieval and Video Recognition. Extensive experiments demonstrate the superiority of our model over the state-of-the-art methods on various datasets, including MSR-VTT, DiDeMo, LSMDC, MSVD, Kinetics-400, and Something-Something-V2. Codes will be available at https://github.com/farewellthree/STAN

1. Introduction

Recent years have witnessed the great success of image-text pretrained models such as *CLIP* [31]. Pretrained on over 400M image-text data pairs, these models learned transferable rich knowledge for various image understanding tasks. Similarly, video domains also call for a *CLIP*-like model to solve downstream video tasks. However, it is hard to get a pretrained model as powerful as *CLIP* in the video domain due to the unaffordable demands on computation resources and the difficulty of collecting video-text data pairs as large and diverse as image-text data. Instead of directly pursuing video-text pretrained models [16, 26], a potential alternative solution that benefits video downstream tasks is to transfer the knowledge in image-text pretrained models to the video domain, which has attracted increasing attention in recent years [11, 12, 25, 28, 29, 40].

Extending pretrained 2D image models to the video domain is a widely-studied topic in deep learning [4, 7], and the key point lies in empowering 2D models with the capability of modeling temporal dependency between video frames while taking advantages of knowledge in the pretrained models. In this paper, based on *CLIP* [31], we revisit temporal modeling in the context of image-to-video knowledge transferring, and present Spatial-Temporal Auxiliary Network (STAN) – a new temporal modeling method that is easy and effective for extending image-text pretrained model to diverse downstream video tasks.

We find that current efforts on empowering *CLIP* with temporal modeling capability can be roughly divided into posterior structure based methods and intermediate structure based methods as shown in Fig. 1(a). Posterior structure based methods [11,12,25] employ a late modeling strategy, which take *CLIP* as a feature extractor and conduct temporal modeling upon the embeddings of video frames extracted independently from *CLIP*. Upon the highly semantic embeddings, though the structure is beneficial for transferring the well-aligned visual-language representation (*i.e.,* high-level knowledge) to downstream tasks, it hardly captures the spatial-temporal visual patterns (*i.e.,* low-level knowledge) among different frames, which is important for video understanding. As shown in Fig. 1(b), compared to the *CLIP* baseline that employs a naive mean pooling to aggregate the features of all frames to obtain a video representation, the performance improvement brought by the typical posterior structure, *i.e.* CLIP4clip-seqTrans [25] is trivial, especially on the video action recognition task where

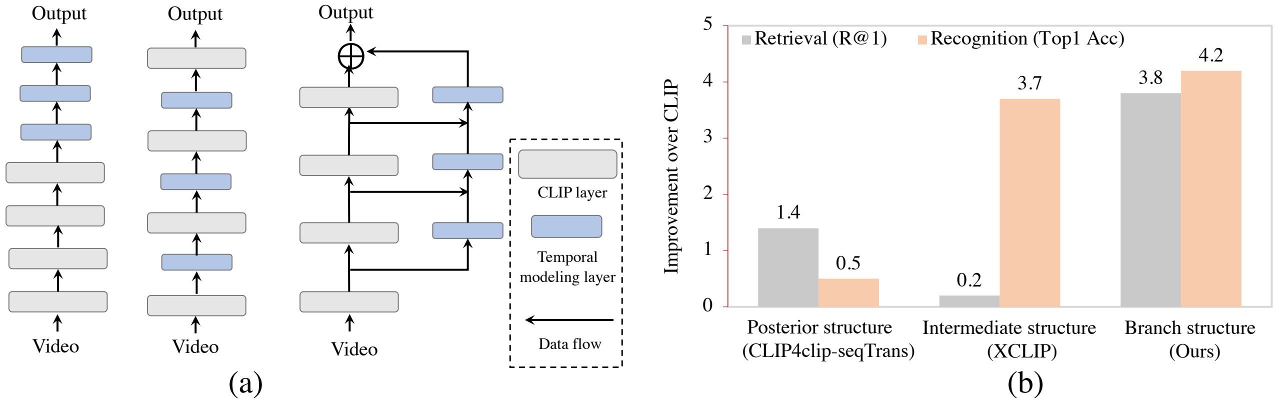

Figure 1. (a) Illustration of temporal modeling with posterior structure (left), intermediate structure (middle) and our branch structure(right). (b) Performance comparison among the posterior structure based CLIP4clip-seqTrans [25] , intermediate structure based XCLIP [28] and our branch structure based STAN. We take the *CLIP* model with a naive mean pooling to aggregate the features of all frames into video representations as the baseline. We present the improvement brought by different methods over this baseline w.r.t. Recall@1 on MSRVTT for video-text retrieval and Top-1 accuracy on Kinetics-400 for video recognition.

spatial-temporal visual patterns are crucial.

In contrast to posterior structure based methods, intermediate structure based methods [4, 28, 29] strengthen the spatial-temporal modeling capability of *CLIP* via plugging temporal modeling modules directly between *CLIP* layers, and achieve 3.7% improvement over the baseline on the video action recognition task. However, we find that inserting additional modules into *CLIP* would impact the pretrained high-level semantic knowledge in the model, which only outperforms the baseline by 0.2% on the video-text retrieval tasks. Therefore, modeling temporal dependency while taking advantage of knowledge in different levels of representation is important for extending the *CLIP* model to the video domain.

Unlike the above methods, inspired by FPN [22] that introduces a branch network to strengthen multi-level representation learning for CNNs, our proposed STAN employs a new branch structure outside of the visual backbone, as shown in Fig. 1(a). Thanks to the branch structure, STAN augments the features of video frames with spatial-temporal contexts at different *CLIP* output levels without affecting the forward-propagating of *CLIP* itself. Thus, it is able to take advantage of both high-level and low-level knowledge in the pretrained model simultaneously, and effectively extends *CLIP* to diverse downstream video tasks. STAN consists of multiple layers with a spatial-temporal separated design. Specifically, the layer operates spatial-temporal modeling via alternatively stacking two separate modules – an intra-frame module and a cross-frame module, which enables the layer to boost the performance of model via reusing the pretrained parameter of *CLIP* layers to initialize the intra-frame spatial modules. We further investigate two instantiations of cross-frame modules, *i.e.,* the self-attention-based module and 3D convolution based module, to facilitate the comprehensive understanding of STAN in different implementations.

We evaluate our STAN on both the high-level semantic-dominant task (*i.e.,* video-text retrieval) and low-level visual pattern-dominant task (*i.e.,*, video recognition), trialing our methods from the two different perspectives. Extensive experiments demonstrate our expanded models are generally effective on the two different tasks. For video-text retrieval, we surpass the CLIP4clip by +3.7%, +3.1%, and +2.1% R@1 on MSRVTT, DiDemo, and LSMDC. For video recognition, we achieve competitive performance on Kinetics-400, with 88× fewer FLOPs than Swin3D-L [24] and improve *CLIP* baseline by 20%+ on Something-Something-V2.

Our main contributions are summarized as: (1) we revisit temporal modeling in the context of image-to-video knowledge transferring and figure out that the key challenge lies in modeling temporal dependency while taking advantage of both high-level and low-level knowledge; (2) we propose Spatial-Temporal Auxiliary Network (STAN) – a new branch structure for temporal modeling, which facilitates representation learning of video frames with including spatial-temporal contexts at different levels and better transfer the pretrained knowledge in *CLIP* to diverse video tasks; (3) our method achieves competitive results on both video-text retrieval and video recognition tasks compared to SOTA methods.

2. Related Work

Visual-Language Pre-Training. Visual-Language pretraining has drawn growing attention in past years [26, 35, 36]. Recently, the contrastive language-image pretraining on web-scale data [17, 31, 45] achieves great success for its remarkable performance when transferring to various downstream tasks. One of the most famous works is the *CLIP* [31], which has revealed surprising capacities of zeroshot recognition and domain generalization [25, 30, 48]. However, language-video datasets suffer from either finite scale [3] or noisy subtitle annotations [26, 44] as well as expensive computation consumes, hence the limited improvement from the pretraining. Thereby, efforts are made [11, 12, 23, 25, 27–29, 41] to adapt the language-image pretraining models to video tasks, which even get better results than methods pretrained on video datasets.

***CLIP* for Video-Text Retrieval.** *CLIP* contains rich vision-text aligned knowledge, which is favoured by the video-text retrieval task. Early works [11, 12, 14, 25, 47] try to add temporal modeling modules as a posterior structure to *CLIP*, *e.g.*, the sequential transformer in [25] and the temporal difference transformer in [11]. Despite the progress they have made, the temporal modeling is limited in high-level embeddings and not effective enough as shown in Fig. 1(b). There are also some works that modify *CLIP* from the perspective of disentangling and multi-level representation interaction [14, 27, 41], and achieve general advancement on various video-text retrieval datasets. However, these methods can only be applied to tasks with sentence input (*i.e.*, multimodal tasks), and are not suitable for recognition tasks. In contrast, our method advances the retrieval as well as other video tasks through effective temporal modeling.

***CLIP* for Video Recognition.** Compared to the retrieval task, the recognition task requires a model to better modeling the dynamic visual patterns in videos, where the visual patterns in *CLIP* learnt from large-scale image-text pretraining data are valuable. Therefore, there are numbers of works migrating the *CLIP* to video recognition [3, 18, 28, 29, 40]. Some of them focus on the prompting or sampling modeling [3, 18, 28], and others [4, 28, 29, 40] design temporal modules as a intermediate structure illustrated in Fig 1(a). Ni *et al* [28] insert the message token to input frame tokens to capture sequence information. Pan *et al* [29] develop 3D convolution modules as adapters plugged between the *CLIP* layers. Unlike the aforementioned methods, we propose a branch structure based method for better transferring image-text model to the video domain.

3. Methodology

3.1. Motivation and Overview

CLIP is a large-scale image-text pretrained model which learns general multi-modal knowledge from 400 million image-text pairs. It consists of two encoders for the extraction of image and text representation respectively, where the visual encoder is composed of a stack of transformer-based [38] encoder layers. From the bottom to the top of layers, the visual encoder gradually learns the visual patterns at different levels of abstraction [46], and at last outputs high-level visual embedding semantically aligned with the corresponding embedding in the text modality.

CLIP-based image-to-video transferring aims to improve the learning of video representation with the pretrained knowledge in *CLIP*, where the key point lies in empowering the image encoder in *CLIP* with the capability of modeling temporal dependency between video frames. Current works typically introduce extra modules as a posterior or intermediate structure of *CLIP* visual encoder for explicitly temporal modeling towards different downstream video tasks. For high-level semantic knowledge dominant tasks, *e.g.,* video-text retrieval, the posterior structure takes advantage of the pretrained visual-language alignment knowledge via operating temporal modeling upon the outputs of *CLIP*. As for visual pattern dominant tasks, *e.g.,* video action recognition, the intermediate structure benefits from the pretrained visual patterns knowledge in *CLIP*, named as low-level knowledge, and empowers the encoder with the capability of learning spatial-temporal patterns from the video. Nevertheless, the posterior structure and the intermediate structure based temporal modeling methods fail to transfer the high-level and low-level knowledge to the video domain simultaneously.

Therefore, we propose Spatial-Temporal Auxiliary Network (STAN), a new temporal modeling mechanism for *CLIP*-based image-to-video knowledge transferring. As shown in Fig. 2, STAN consists of a stack of K spatial-temporal layers and acts as a branch structure beside the *CLIP* visual encoder. Given a video with T frames, the frames are fed into the *CLIP* visual backbone to obtain intermediate outputs at last $K+1$ level of *CLIP* layers. We denote the outputs of the kth selected *CLIP* layer as:

$$V^k = \{f^k_{i,j} \in \mathcal{R}^D | i \in [1,T], j \in [0,L]\}, \tag{1}$$

which is a visual embedding sequence of the video where T, L and D are the frame number, per-frame patch number and embedding dimension, respectively. In V^k, $f^k_{i,0}$ indicates the embedding of [CLS] token in the ith frame of the video while $f^k_{i,j>0}$ represents the visual embedding of jth patch in the frame. Then, we feed each intermediate output V^k into the corresponding level of layer in STAN for the modeling of spatial-temporal correspondence between video frames. At last, we fuse the output of the last *CLIP* layer with the output of STAN to get the final representation of the video.

Compared to the posterior structure based methods, STAN operates spatial-temporal modeling on multi-level *CLIP* representations and thereby is able to better capture the visual dynamics information in the video. Meanwhile, unlike previous intermediate structure based methods, which insert additional modules into *CLIP* visual encoder, the branch structure of STAN avoids destroying the inherent structure of the visual encoder and thereby protect the pretrained knowledge, especially the high-level visual-text alignment knowledge in *CLIP*.

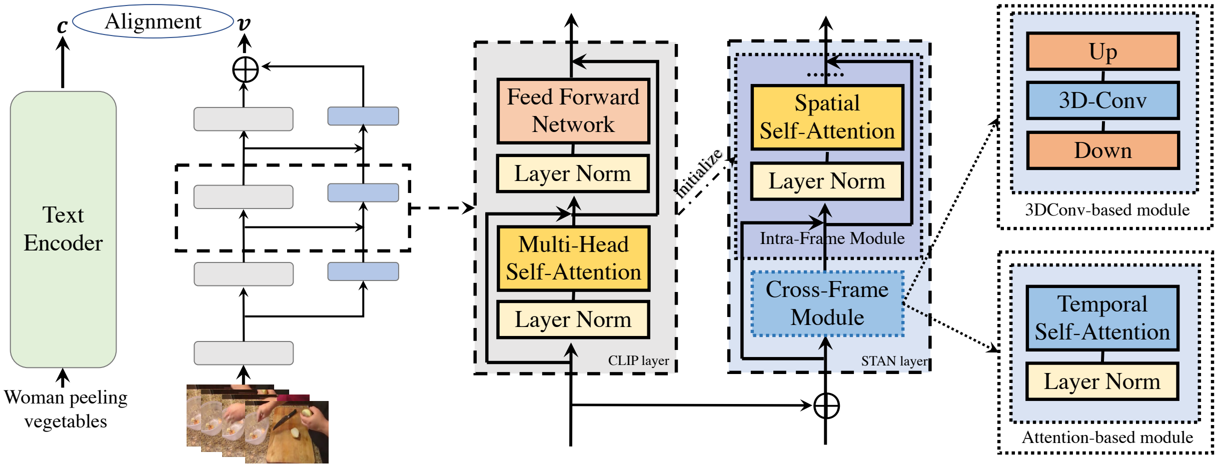

Figure 2. The overview of our proposed STAN architecture, including the global overview of our backbone (left), details of the internal structure of our spatial-temporal module (middle), and implementations of the cross-frame module (right).

3.2. Spatial-Temporal Auxiliary Network

STAN consists of a stack of K spatial-temporal layers, where the input for each layer is constructed based on the output of a *CLIP* visual layer. For the kth layer in STAN, its input is an embedding sequence of the video denoted as:

$$V'^k = \{f'^k_{0,0}, f'^k_{1,1}, .., f'^k_{1,L}, .., f'^k_{T,1}, .., f'^k_{T,L}\}, \tag{2}$$

where $f'^k_{0,0}$ is the embedding of [CLS] token for the whole video while others are embedding of image patches in different frames. The output of the STAN layer is also an embedding sequence with the same size as its input, which is denoted as:

$$\widetilde{V}^k = \{\widetilde{f}^k_{0,0}, \widetilde{f}^k_{1,1}, .., \widetilde{f}^k_{1,L}, .., \widetilde{f}^k_{T,1}, .., \widetilde{f}^k_{T,L}\}, \tag{3}$$

At the first STAN layer, to construct its input from V^1, we first average the embedding of [CLS] tokens in each frame as a new embedding $f'^1_{0,0} = \frac{1}{T}\sum_{i \in T} f^1_{i,0}$, and then update patch embeddings in V^1 with spatial and temporal position embeddings as:

$$f'^1_{i,j} = \mathrm{Dropout}(f^1_{i,j} + \mathrm{Pos_t}(t) + \mathrm{Pos_s}(j)), \tag{4}$$

where $j > 0$ while $\mathrm{Pos_t}$ and $\mathrm{Pos_s}$ are the learnable embeddings for the temporal and spatial position of each patch. For the other layers in STAN, the input V'^k is constructed based on the output from the previous STAN layer $\widetilde{V}^{k-1}$ and *CLIP* output V^k as follows:

$$f'^k_{0,0} = \widetilde{f}^{k-1}_{0,0} + \mathrm{W}^k_{proj} \frac{1}{T} \sum_{i \in T} f^k_{i,0}, \tag{5}$$

$$f'^k_{i,j} = \widetilde{f}^{k-1}_{i,j} + \mathrm{W}^k_{proj} f^k_{i,j}, \tag{6}$$

where $i \in [1, T], j \in [1, L]$, and $\mathrm{W}^k_{proj} \in \mathbf{R}^{D \times D}$ is a projection layer.

Given the input embedding sequence of the video, STAN layer learns the spatial-temporal information among the video frames. As shown in Fig. 2, it operates temporal modeling via alternatively stacking two separated modules – an intra-frame module and a cross-frame module. Thanks to the separated design, we are able to reuse the structure in *CLIP* visual encoder layer as our intra-frame spatial module and initialize it with the pretrained model, which effectively improves the performance on downstream tasks. Same as *CLIP*, the intra-frame module is also a self-attention block responsible for spatial modeling. For simplicity, we omit the superscript of embedding and denote the embeddings in frame i as $X_i \in \mathbf{R}^{(L+1) \times D}$, where the embedding of [CLS] token in the video is duplicated and concatenated with patch embeddings. In each frame, the spatial module updates embeddings via self-attention:

$$\hat{X}_i = \mathrm{softmax}(X_i \mathrm{W_Q}(X_i \mathrm{W_K})^{\mathrm{T}} / \sqrt{D})(X_i \mathrm{W_V}) + X_i, \tag{7}$$

where $\mathrm{W_Q}/\mathrm{W_K}/\mathrm{W_V}$ indicate the linear projections for the query, key and value in self-attention layer of the spatial module. After that, the duplicated [CLS] embeddings in each frame are averaged as the video [CLS] embeddings.

The cross-frame module is responsible for temporal modeling. For simplicity, we omit the superscript of embedding and denote the collection of jth patch embeddings in different frames as $Y_j \in \mathbf{R}^{T \times D}$. At each spatial position, the patch embeddings are updated as $\hat{Y}_j = Temp(Y_j)$, where $Temp()$ indicates the message passing strategy across temporal dimension which can be instantiated in different ways. In the next section, we present a self-attention-based cross-frame module and a 3D convolution-based cross-frame module, and study the performance of the two instantiations.

3.3. Temporal Modeling in STAN

In deep learning, there are various ways to achieve temporal modeling, for example, 3D convolution [7,37], temporal self-attention [2,4] and proxy tokens [10,28]. In this paper, we investigate two most popular instantiations of temporal modeling in the proposed framework, *i.e.,* the self-attention based module and convolution based module, to facilitate the comprehensive understanding of STAN in different implementations.

Self-attention based module. Self-attention has a natural advantage in sequence modeling due to its global modeling capability. At each spatial position, the patch embeddings from different frames are updated as:

$$\hat{Y}_i = \mathrm{softmax}(Y_i \mathrm{W_Q}(Y_i \mathrm{W_K})^{\mathrm{T}}/\sqrt{D})(Y_i \mathrm{W_V}) + Y_i, \quad (8)$$

where $\mathrm{W_Q}/\mathrm{W_K}/\mathrm{W_V}$ indicate the linear projections for the query, key, and value in self-attention layer of the cross-frame module. Through temporal attention, each patch is contextualized with temporal information at the same locations.

Convolution based module. Convolution operator has been widely adapted for effective temporal modeling in CNNs [7, 37, 42], e.g., C3D [37], S3D [42]. Though self-attention gains increasing attention, convolution still owns the advantage of better local modeling and easier easier to converge. Therefore, we also implement the cross-frame module of STAN based on the convolution operator. Specifically, we stack the patch embeddings of the video to form a 3D feature cube $Y \in \mathbf{R}^{T\times W\times H\times D}$ and then update the features as follows:

$$Y = Up(\mathrm{Gelu}(\mathrm{3DConv}(Down(Y)))) + Y, \quad (9)$$

where the $Down()$ and $Up()$ are the point-wise convolution operators with channel size of $\frac{D}{8}$ and D, which reduce and restore the dimension of patch embeddings. As for the kernel size of 3D convolution, the dimensions for T, H, and W are set to 3, 1, and 1 respectively.

4. Experiments

4.1. Experiment Settings

Datasets. We evaluate our method on both the high-level semantic-dominant task *i.e.,*, video-text retrieval, and low-level visual pattern-dominant task *i.e.,*, video recognition, trialing our methods from the two different perspectives. For video-text retrieval, we employ *MSR-VTT* [43], *DiDemo* [1] and *LSMDC* [32]; for video recognition, we adopt *Kinetics-400* [19] and *Something-Something-v2* [15].

MSR-VTT is the most popular benchmark consisting of 10,000 YouTube videos with 20 captions for each video. *DiDemo* contains 10,000 videos and 40,000 sentences with longer video duration than other retrieval datasets. *LSMDC* is a large-scale video-text retrieval benchmark with 118,081 videos from 202 movies, which is more diverse in concept and duration than other datasets.

Kinetics-400 (K-400) is a popular video action recognition dataset that has 260,000 videos with average 300 frames and 400 action classes. *Something-Something-v2 (SSv2)* is a video action recognition benchmark especially for temporal modeling, which contains 220,485 videos and 174 action classes. In *K-400*, most of the action categories are biased to static scene context [34]. In *SSv2*, the classes of action are less relevant to the static scene context but closely related to the dynamic information in videos.

Implementation Details. We set the number of STAN layers to 4 for all datasets except on SSv2 when it is set to 6. We employ the simple cross-entropy loss and NCE loss for fine-tuning on video recognition and video-text retrieval, respectively. Following previous work [25], we fine-tune the model with a frame number of 12 and a token length 32 for MSRVTT, LSMDC, K400, and SSv2. On Didemo where videos have a longer duration, the frame number and token number are set to 64 and 64. The batch size is set to 128 for all datasets. We adopt Adam as our optimizer with weight decay of 0.02. The learning rates are initialized to 2e-6 and 2e-5 for parameters in *CLIP* and parameters in STAN respectively, and then decay following a cosine annealing decay schedule. For more details and code, please refer to supplementary materials.

4.2. Comparisons with State-of-the-art

Video-Text Retrieval. We compare our STAN with current SOTAs including both video-text pretrained and *CLIP*-pretrained methods across different benchmarks. Comparisons on MSR-VTT, DiDemo abd LSMDC are reported in Table 1, 2, and 3, respectively. For *CLIP*-pretrained methods, unless denoted with *B/16*, all the methods are based on *CLIP*-B/32. For our method, we report the results achieved by both *CLIP*-B/32 and *CLIP*-B/16, and denote STAN with self-attention and 3D convolution based inter-frame module as STAN-self and STAN-conv, respectively.

As shown in Table 1, 2 and 3, *CLIP*-based methods generally achieve superior performance than the video-text pretrained methods, which demonstrates the great potential of transferring image-text pretrained models to the video domain. Among the *CLIP*-based methods, our STAN achieves SOTA performance across all three benchmarks at both *CLIP*-B/32 and *CLIP*-B/16 model scales. Specifically, with comparable model size, STAN outperforms the posterior structure based method, *i.e.,* CLIP4clip [25] by 2.9% at R@1 averaged on the three datasets, which shows obvious advantage of the branch structure. Compared to the other SOTAs, *e.g.,* DRL [41], which advances video-text retrieval via improving the cross-modality interaction upon visual-language outputs of *CLIP*, STAN shows a different way to

Table 1. Comparisons on MSR-VTT [43]. We train on Training-9K and test on Test-1k-A. * means extra tricks (*e.g.,* DSL [8] and QB-Norm [5]) are utilized during inference.

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓
Pretrained on large-scale video-text dataset				
ClipBERT [20]	22.0	46.8	59.9	6.0
Frozen [3]	31.0	59.5	70.5	3.0
HD-VILA [44]	35.6	65.3	78.0	3.0
All-in-one [39]	37.9	68.1	77.1	-
BridgeFormer [13]	37.6	64.8	75.1	3.0
Clover [16]	38.6	67.4	76.4	2.0
CLIP pretrained				
Clip4clip [25]	44.5	71.4	81.6	2.0
CenterCLIP [47]	44.2	71.6	82.1	2.0
CLIP2Video* [11]	47.2	73.0	83.0	-
CAMoE* [8]	47.3	74.2	84.5	3.0
CLIP2TV-B/16 [12]	49.3	74.7	83.6	2.0
DRL-B/16* [41]	53.3	**80.3**	87.6	1.0
Our method				
STAN-self-B/32	46.9	72.8	82.8	2.0
STAN-conv-B/32	46.6	72.8	82.2	2.0
STAN-self-B/32*	49.0	74.8	83.5	2.0
STAN-self-B/16	50.0	75.2	84.1	1.5
STAN-self-B/16*	**54.1**	79.5	**87.8**	1.0

Table 2. Comparisons on DiDemo [1]. We concatenate all captions of a video into a single query. * means extra tricks (*e.g.,* DSL [8] and QB-Norm [5]) are utilized during inference.

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓
Pretrained on large-scale video-text dataset				
ClipBERT [20]	20.4	48.0	60.8	6.0
Frozen [3]	31.0	59.8	72.4	3.0
HD-VILA [44]	28.8	57.4	69.1	4.0
All-in-one [39]	32.7	61.4	73.5	3.0
BridgeFormer [13]	37.0	62.2	73.9	3.0
Clover [16]	48.6	74.3	82.2	2.0
CLIP pretrained				
Clip4clip [25]	43.4	70.2	80.6	2.0
CAMoE* [8]	43.8	71.4	-	-
CLIP2TV [12]	45.5	69.7	80.6	2.0
DRL-B/16 [41]	49.0	76.5	84.5	2.0
Our method				
STAN-self-B/32	46.2	70.4	80.0	2.0
STAN-conv-B/32	46.5	71.5	80.9	2.0
STAN-conv-B/32*	51.3	75.1	83.4	1.0
STAN-conv-B/16	49.4	74.9	84.5	1.0
STAN-conv-B/16*	**54.6**	**78.4**	**85.1**	1.0

Table 3. Comparison on LSMDC [32]. * means extra tricks (*e.g.,* DSL [8] and QB-Norm [5]) are utilized during inference.

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓
Pretrained on large-scale video-text dataset				
Frozen [3]	15.0	30.8	40.3	20.0
HD-VILA [44]	17.4	34.1	44.1	15.0
BridgeFormer [13]	17.9	35.4	44.5	15.0
Clover [16]	22.7	42.0	52.6	9.0
CLIP pretrained				
Clip4Clip [25]	21.6	41.8	49.8	8.0
CAMoE* [8]	25.9	46.1	53.7	-
CCLIP-B/16 [47]	24.2	46.2	55.9	8.0
DRL-B/16 [41]	26.5	47.6	56.8	7.0
Our method				
STAN-self-B/32	23.7	42.7	51.8	9.0
STAN-conv-B/32	23.1	42.2	51.0	9.0
STAN-self-B/32*	26.2	46.0	53.9	9.0
STAN-self-B/16	27.1	49.3	58.7	6.0
STAN-self-B/16*	**29.2**	**49.5**	**58.8**	6.0

achieve competitive performance, which improves the temporal modeling capability of *CLIP* itself. Therefore, empowering *CLIP* model with stronger video representation learning capability, STAN is potentially compatible with the other SOTAs which present advanced techniques operated upon *CLIP* outputs, e.g., hierarchical video-text interaction [27, 41] and hard sample modeling [11]. We leave them for future work. Additionally, we also notice that both the self-attention and 3D convolution instantiated model, *i.e.,* STAN-self and STAN-conv, achieve competitive performance with a slight difference. Specifically, STAN-conv is comparable with STAN-self when transferring to smaller datasets, e.g., MSRVTT (-0.3 at R@1) and DiDeMo (+0.3 at R@1) while STAN-self is better on larger scale dataset, e,g., LSMDC (+0.6 at R@1). The results further suggest that self-attention instantiated STAN would be the better choice when transferring *CLIP* to large-scale downstream datasets, while 3D convolution instantiated STAN would be better for the small ones. In Appendix, we present more results with visualization.

Video Recognition. To evaluate the spatial-temporal modeling capability of STAN, we compare it to other SOTAs on video recognition benchmarks, *i.e.,* Kinetics-400 (K400) and Something-Something-v2 (SSv2). The results are reported in Table 4 and Table 5 respectively. On K400 benchmark, *CLIP*-based methods achieve competitive results with much smaller model size compared to the image-pretrained methods, which shows the superiority of image-text pretraining. For example, our VIT-B/16 based STAN outperforms VIT-Huge based ViViT [2] and Swin3D-L based Video-swin [24], which have more than 15× and 88× GFLOPs compared to our method. Meanwhile, our method achieves SOTA performance among *CLIP*-based methods, which demonstrates the effective of our method on transferring *CLIP* to the video domain. As for SSv2 benchmark, we find that, without temporal modeling, bare *CLIP* model [9] achieves only 44.0% top-1 accuracy which dramatically under-performs ImageNet-21K pretrained Timesformer [6], though it owns pretrained knowledge obtained from a much larger image-text dataset. The result suggest that the domain gap is significant between SSv2 and *CLIP* model, and temporal modeling capability is desired for the action recognition task on SSv2. STAN brings about more than 20% performance improvement over the *CLIP* baseline and achieves competitive compared to other *CLIP*-based methods, which demonstrates that STAN empowers *CLIP* with strong temporal modeling capability.

4.3. Ablation Study

To verify the contribution of different components in our method, we conduct ablation experiments on both video-text retrieval tasks (*i.e.,* MSR-VTT and DiDemo) and video action recognition tasks (*i.e.,* K400 and SSv2). First of all, according to the results reported in Table 6, we can conclude that components in STAN are compatible with each

Table 4. Comparison between our method and the state-of-the-arts on Kinetics-400 validation set [19]. We report the FLOPs of all views.

Methods	Pretrain	Frames	Testing Views	GFLOPs	Top-1 Accuracy	Top-5 Accuracy
Large-scale image pretraining						
TimeSformer-L [4]	ImageNet-21 K	96	1×3	7140	80.7	94.7
Video-Swin-L (384 ↑) [24]	ImageNet-21 K	32	10×5	105350	84.9	96.7
MViTv2-L (312 ↑) [21]	ImageNet-21 K	40	5×3	42420	**86.1**	**97.0**
ViViT-H [2]	JFT-300M	32	4×3	17352	84.8	95.8
TokenLearner-L/10 [33]	JFT-300M	-	4×3	48912	85.4	96.3
Large-scale image-text pretraining						
CLIP-B/16 [9]	CLIP-400M	8	4×3	-	81.1	94.8
Action-CLIP-B/16 [40]	CLIP-400M	32	10×3	16890	83.8	96.2
A6 [18]	CLIP-400M	16	–	-	76.9	93.5
STadapter-CLIP-B/16 [29]	CLIP-400M	8	1×3	455	82.0	95.7
STadapter-CLIP-B/16 [29]	CLIP-400M	32	1×3	1821	82.7	96.2
X-CLIP-B/16 [28]	CLIP-400M	8	4×3	1740	83.8	96.7
X-CLIP-B/16 [28]	CLIP-400M	16	4×3	3444	84.7	**96.8**
Our method						
STAN-conv-B/16	CLIP-400M	8	1×3	714	83.1	96.0
STAN-self-B/16	CLIP-400M	8	1×3	593	84.2	96.5
STAN-self-B/16	CLIP-400M	16	1×3	1187	**84.9**	**96.8**

Table 5. Comparison on Something-Something-v2 validation set [15]. We report the FLOPs of all views. * means our implementation.

Methods	Pretrain	Frames	Testing Views	GFLOPs	Top-1 Accuracy	Top-5 Accuracy
TimeSformer-HR [4]	ImageNet-21 K	16	1×3	5109	62.5	-
ViViT-L [2]	K400	16	4×3	11892	65.4	89.8
MViT-B-24 [21]	K600	32	1×3	708	68.7	91.5
Video-Swin-B [24]	K400	32	1×3	963	**69.6**	**92.7**
CLIP-B/16 [9]	CLIP-400M	8	1×3	-	44.0	76.2
X-CLIP-B/16* [40]	CLIP-400M	8	1×3	435	63.1	89.0
STadapter-CLIP-B/16 [40]	CLIP-400M	8	1×3	489	67.1	91.2
STadapter-CLIP-B/16 [40]	CLIP-400M	32	1×3	1955	69.5	92.6
Our method						
STAN-conv-B/16	CLIP-400M	8	1×3	845	65.2	90.5
STAN-self-B/16	CLIP-400M	8	1×3	688	67.6	91.4
STAN-self-B/16	CLIP-400M	16	1×3	1376	69.5	**92.7**

other while each of them contributes to the transferring of *CLIP*. Specifically, when we remove the branch structure and multi-level feature learning, and append STAN as a posterior structure upon the *CLIP*, the performance of STAN decreased a lot on all four benchmarks, which demonstrates the superiority of our model structure compared to the posterior structure. Besides, we find that without the Cross-Frame module, STAN still brings about performance improvement over baseline, which suggests that our method is beneficial to image-to-video knowledge transferring for *CLIP* model. With the help of Cross-Frame module, the complete STAN further outperforms the baseline by a larger margin and achieves SOTA performance on both video-text retrieval and video recognition tasks, which reveals our method is able to model temporal dependency while taking advantage of knowledge in different level of representation.

4.4. Further discussion on STAN

The effect of different temporal modeling structures. As aforementioned, posterior structure based temporal modeling for *CLIP* transferring is popular for high-level knowledge dominant tasks, *e.g.,* video-text retrieval, while intermediate structures are employed for low-level knowledge dominant tasks, *e.g.,* video recognition. In section 4.2, we demonstrate the superiority of our branch structured based method compared to other structures on different tasks, respectively. Here, we further adopt posterior structures for video recognition and intermediate structure for video-text retrieval to better understand the effect of different temporal modeling structures. For posterior structures, we employ the Sequential Transformer (Seq Trans) in CLIP4clip [25] and Temporal Differential Block (TDB) in CLIP2video [11]. For intermediate structures, we choose the Message Token (Msg Token) in XCLIP [28] and ST-adapter [29]. Note that, for a fair comparison, we only report the performance achieved by the temporal modeling structures without other extra techniques (*e.g.*, prompting modeling in [11] and [28]). As shown in Table 7, posterior structures are more effective than intermediate structures in transferring *CLIP* to video-text retrieval tasks, but brings trivial improvement on the video recognition task. In contrast, intermediate structures perform well on video recognition, but bring little improvement over baseline on video-text retrieval. As for our branch structure based STAN, it

Table 6. Ablation studies on different datasets. For MSRVTT and DiDemo, we use *CLIP*-B/32 as backbone and report Recall@1; for K400 and SSv2, we use *CLIP*-B/16 as backbone and report Top1 Accuracy. We adopt temporal self-attention here in our Cross-Frame module.

Components				Results			
Cross-Frame	Intra-Frame	Branch structure	Multi-level	MSR-VTT	DiDemo	K400	SSv2
				43.1	43.4	79.9	44
✓	✓			44.9	43.5	80.5	55.9
✓	✓	✓		44.2	43.6	80.8	58.6
	✓	✓	✓	44.3	44.5	81.0	48.1
✓		✓	✓	43.1	43.7	80.0	55.7
✓	✓	✓	✓	**46.9**	**46.2**	**82.6**	**65.9**
+ Testing Techniques (DSL [8] or 1 × 3-views)				49.7	51.4	84.2	67.6

Table 7. Analysis of different structures on both video recognition and retrieval tasks. * means our own implementation.

Method	Style	MSR-VTT	K400
baseline	-	43.1	79.9
Msg Token [28]	Intermediate	43.2*	82.7
ST-adapter [29]	Intermediate	42.5*	82.0
Seq Trans [25]	Posterior	44.5	80.5*
TDB [11]	Posterior	45.1	81.1*
STAN-selfN	Branch	**46.9**	**84.2**

Table 8. The impact of different levels of inputs from *CLIP* layers on STAN.

	Methods	Recall@1 on MSR-VTT	Top1 Acc on SSv2
	Baseline	43.1	44.0
Interval	3	43.5	54.4
	2	44.2	61.1
	1	**46.9**	**65.2**
Range	1-4	43.4	62.2
	5-8	43.9	62.3
	9-12	**46.9**	**65.2**

not only successfully extends *CLIP* to both tasks but also outperforms both the other two structure based methods, which demonstrates that our structure is a better temporal modeling method in the context of *CLIP*-based image-to-video knowledge transferring.

The impact of multi-level inputs from *CLIP* layers. STAN acts as a new branch beside the *CLIP* backbone, which takes the video frame representation at different levels of *CLIP* layers as inputs. To study the impact of the choice of different *CLIP* representations, we fixed the number of STAN layers to 4 and vary the level range and interval of selected *CLIP* layers. For the interval, we align the last layer of *CLIP* and STAN, and vary the interval between the selected *CLIP* layers. For example, interval=2 means STAN receives outputs of every 2 *CLIP* layers as inputs, *i.e.,* the $6th$, $8th$, $10th$, and $12th$ layers. As shown in Table 8, interval=1 is the best choice for both datasets. Then, we fix the interval to 1, and vary the level range of selected *CLIP* layers. The result suggests that the mid-high level of pretrained *CLIP* representation is more valuable for downstream tasks.

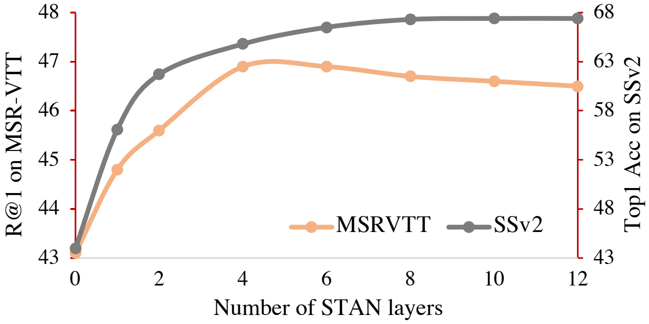

Figure 3. Analysis of the number of layers in STAN. We report the Recall@1 on MSR-VTT and Top1 Accuracy on SSv2 respectively.

The impact of STAN layer number. Intuitively, increasing layers in STAN would access more *CLIP* representation levels bringing about stronger temporal modeling capability. Nevertheless, it also increases the risk of over-fitting to downstream tasks. Thereby, we study the effect of layer number to find a better trade-off. As shown in Fig. 3, for MSR-VTT retrieval, the performance improvement of STAN reaches the peak at 4 layers, and the performance drops with further increasing of layers. On SSv2, the performance improvement of STAN tend to converge after 6 layers. Generally, STAN with 4 to 6 layers is a good choice for different tasks.

5. Conclusion

In this paper, we study the temporal modeling in *CLIP*-based image-to-video knowledge transferring. We first uncover that current methods fail to work on high-level semantic-dominant tasks and low-level visual pattern-dominant tasks simultaneously. Then, to address this problem, we propose a simple and effective framework named Spatial-Temporal Auxiliary Network (STAN) to expand *CLIP* to diverse video tasks. Extensive experiments on Video-Text Retrieval and Video Recognition tasks demonstrate the superiority of our method.

Acknowledgements. This work was supported by the National Key R&D Program of China (No. 2020AAA0103501), Shenzhen Fundamental Research Program (GXWD20201231165807007 - 20200806163656003) and National Natural Science Foundation of China (No. 62172021).

*: equal contribution

◇: Work done when interning at ByteDance Inc

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017. 5, 6

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 5, 6, 7

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 3, 6

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 1, 2, 3, 5, 7

[5] Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. Cross modal retrieval with querybank normalisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5194–5205, 2022. 6

[6] Adrian Bulat, Juan Manuel Perez Rua, Swathikiran Sudhakaran, Brais Martinez, and Georgios Tzimiropoulos. Space-time mixing attention for video transformer. *Advances in Neural Information Processing Systems*, 34:19594–19607, 2021. 6

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 5

[8] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021. 6, 8

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6, 7

[10] Quanfu Fan, Chun-Fu Chen, and Rameswar Panda. Can an image classifier suffice for action recognition? In *International Conference on Learning Representations*, 2021. 5

[11] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021. 1, 3, 6, 7, 8

[12] Zijian Gao, Jingyu Liu, Sheng Chen, Dedan Chang, Hao Zhang, and Jinwei Yuan. Clip2tv: An empirical study on transformer-based methods for video-text retrieval. *arXiv preprint arXiv:2111.05610*, 2021. 1, 3, 6

[13] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridging video-text retrieval with multiple choice questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16167–16176, 2022. 6

[14] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5006–5015, 2022. 3

[15] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 5, 7

[16] Jingjia Huang, Yinan Li, Jiashi Feng, Xiaoshuai Sun, and Rongrong Ji. Clover: Towards a unified video-language alignment and fusion model. *arXiv preprint arXiv:2207.07885*, 2022. 1, 6

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

[18] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. *arXiv preprint arXiv:2112.04478*, 2021. 3, 7

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5, 7

[20] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. 6

[21] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 7

[22] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[23] Ruyang Liu, Hao Liu, Ge Li, Haodi Hou, TingHao Yu, and Tao Yang. Contextual debiasing for visual recognition with causal mechanisms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12755–12765, 2022. 3

[24] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 2, 6, 7

[25] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021. 1, 2, 3, 5, 6, 7, 8

[26] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 1, 2, 3

[27] Shaobo Min, Weijie Kong, Rong-Cheng Tu, Dihong Gong, Chengfei Cai, Wenzhe Zhao, Chenyang Liu, Sixiao Zheng, Hongfa Wang, Zhifeng Li, et al. Hunyuan_tvr for text-video retrivial. *arXiv preprint arXiv:2204.03382*, 2022. 3, 6

[28] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. *arXiv preprint arXiv:2208.02816*, 2022. 1, 2, 3, 5, 7, 8

[29] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning for action recognition. *arXiv preprint arXiv:2206.13559*, 2022. 1, 2, 3, 7, 8

[30] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 3

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2

[32] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017. 5, 6

[33] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Advances in Neural Information Processing Systems*, 34:12786–12797, 2021. 7

[34] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 535–544, January 2021. 5

[35] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 2

[36] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. 2

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 5

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[39] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022. 6

[40] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 1, 3, 7

[41] Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan, and Xian-Sheng Hua. Disentangled representation learning for text-video retrieval. *arXiv preprint arXiv:2203.07111*, 2022. 3, 5, 6

[42] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 5

[43] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 5, 6

[44] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5036–5045, 2022. 3, 6

[45] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2

[46] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021. 3

[47] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. *arXiv preprint arXiv:2205.00823*, 2022. 3, 6

[48] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3