Prompting Visual-Language Models for Efficient Video Understanding

Chen Ju[1], Tengda Han[2], Kunhao Zheng[1], Ya Zhang[1(✉)], and Weidi Xie[1,2(✉)]

[1] Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China
{ju_chen,dyekuu,ya_zhang,weidi}@sjtu.edu.cn,
[2] Visual Geometry Group, University of Oxford, Oxford, England
htd@robots.ox.ac.uk
https://ju-chen.github.io/efficient-prompt

Abstract. Image-based visual-language (I-VL) pre-training has shown great success for learning joint visual-textual representations from large-scale web data, revealing remarkable ability for "zero-shot" generalisation. This paper presents a simple but strong baseline to efficiently adapt the pre-trained I-VL model for video understanding tasks, with minimal training. Specifically, we propose to optimise a few random vectors, termed as "continuous prompt vectors", that convert video-related tasks into the same format as the pre-training objectives. In addition, to bridge the gap between static images and videos, temporal information is encoded with lightweight Transformers stacking on top of frame-wise visual features. Experimentally, we conduct extensive ablation studies to analyse the critical components. On ten public benchmarks of action recognition, action localisation, and text-video retrieval, across closed-set, few-shot, and zero-shot scenarios, we achieve competitive or state-of-the-art performance to existing methods, despite optimising significantly fewer parameters. Due to space limitation, we refer the readers to the arXiv version at https://arxiv.org/abs/2112.04478.

1 Introduction

While the research in computer vision has mainly focused on tackling particular tasks, the grand goal towards human-level perception has always been to learn general-purpose visual representation, that can solve various problems with *minimal tunings*. Towards such a goal, recent work for training image-based visual-language **(I-VL)** models has shown promising progress. For example, CLIP [60] and ALIGN [30] learn the joint representation for image and text with simple noise contrastive learning, greatly benefiting from the rich information in text descriptions, *e.g.* actions, objects, human-object interactions, and object-object relationships. As a result, these pre-trained I-VL models have demonstrated

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19833-5_7.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
S. Avidan et al. (Eds.): ECCV 2022, LNCS 13695, pp. 105–124, 2022.
https://doi.org/10.1007/978-3-031-19833-5_7

remarkable "zero-shot" generalisation for various image classification tasks. Crucially, the data used to train these powerful I-VL models can simply be crawled from the Internet at scale, without any laborious manual annotation. It is therefore reasonable to believe, with the growing computation, larger datasets will be collected, and more powerful models will be trained in the near future.

Given this promise, one question naturally arises: *how can we best exploit the ability in the powerful I-VL models, and effectively adapt it to solve novel vision tasks of interest?* One possible solution would be to finetune the image encoder end-to-end on the downstream tasks, however, since each task needs to finetune and save its own set of parameters, we end up developing hundreds of models for hundreds of individual tasks. Even more problematic, discarding the text encoder loses the model's ability for "zero-shot" generalisation, thus the resultant model can only work for a fixed set of pre-determined categories. Alternatively, as shown in CLIP [60], given properly designed "prompts", the model is able to work on a variety of downstream tasks including "zero-shot", with classifiers being dynamically generated by the text encoder, from category names or other free-form texts. The prompts here are handcrafted close templates to facilitate classifier generation, so that novel tasks can be formulated in the same format as pre-training objectives, effectively closing the gap between pre-training and downstream tasks. One remaining issue is, such handcrafted prompts require extensive expert knowledge and labor, limiting the use for efficient adaptation.

In this paper, we continue the vein of prompt-based learning [39,40], with the goal of exploring a comprehensive and strong baseline to adapt I-VL models for *efficient* video understanding. We here focus on the resource-hungry video tasks, for three reasons: 1) From the *data* perspective, comparing to image-text pairs, video-text pairs are harder to collect, and may suffer from misalignment issues [25]; 2) Solving video tasks demands more computational power. Given the same budget, training on image-text pairs enables the model to learn more *diversity*, making it more cost-effective to understand video with I-VL models; 3) Videos are composed of frame sequences, establishing temporal dependencies on powerful image-based models is a natural choice.

Specifically, we consider a simple idea by prepending/appending a sequence of random vectors, termed as "continuous prompt vectors", to the textual input. These prompt vectors consist entirely of free parameters that do not correspond to any real concrete words, and the subsequent layers of the text encoder will attend these vectors, as if they were a sequence of "virtual tokens" to generate the corresponding classifier or embedding. During training, we freeze the weights of the I-VL text encoder, and the gradients are back-propagated to optimise these learnable prompt vectors. Consequently, a single copy of the visual backbone is able to perform various video tasks, with the minimal number of trainable parameters for each task. To further exploit the video temporal information, we also append *lightweight* Transformers on top of frame-wise visual representation. As a result, the various video tasks can be formulated under the same umbrella, *i.e.* to maximise the similarity matching between visual and textual embeddings, with texts being action category names or fine-grained descriptions.

To summarise, building on scalable and powerful I-VL models, we first propose a simple baseline for *efficient* and *lightweight* video understanding, by learning task-specific prompt vectors, which facilitates possible future research in action recognition, action localisation, and text retrieval; We extensively evaluate on ten public benchmarks, across closed-set, few-shot, and zero-shot scenarios, thoroughly dissect critical components; Lastly, despite training only a few free parameters, *i.e.* several prompt vectors and two Transformer layers, in the closed-set scenario, we achieve competitive or state-of-the-art performance to existing methods. In the few-shot and zero-shot scenarios, we significantly outperform all previous methods on seven popular benchmarks, sometimes by over 10% gains.

2 Related Work

Joint Visual-Textual Learning. In the literature, [57] has explored the connection between images and words using paired text documents, and [17,75] proposed to jointly learn image-text embeddings with the category name annotations. Recently, CLIP [60], ALIGN [30] and FILIP [81] have further scaled up the training with large-scale web data. Using simple noise contrastive learning, it is shown that powerful visual representation can be learnt from paired image-caption. In video domains, similar idea has also been explored for representation learning [52] and video retrieval [1,38,53]. In this paper, we establish baselines on steering the pre-trained CLIP model to video understanding tasks, the same technique should be applicable to other I-VL models as well.

Prompting refers to designing proper "instructions" that the pre-trained language model can understand, and generate desired outputs, using a few examples as demonstrations. Given properly handcrafted prompt templates, GPT-3 [6] has shown strong few-shot or zero-shot generalisations. However, such handcrafted templates require extensive expert knowledge, limiting the flexibility. Later work proposes to automate prompt engineering by searching discrete prompts [22,33,62,63], and continuous prompts [39,40]. This work considers to search continuous prompts for steering I-VL models to tackle various video tasks.

Video Action Recognition. Effective architecture research has gone through rapid developments, from two-stream networks [16,66,71] to more recent single stream RGB networks [3,10,14,15,70,76]. With the help of abundant data [9], recognition accuracy has been steadily improved. In addition, data-efficient learning has been explored: few-shot and zero-shot action recognition. Specifically, few-shot recognition makes only a few training samples available from each action category. [91–93] proposed compound memory networks to match and rank videos; [8,13,59] used GANs, dynamic time warping, and CrossTransformer to synthesize or align examples for novel categories. While zero-shot recognition requires to generalise towards action categories that are unseen during training. One typical idea lies in learning a common representation space shared by seen and unseen categories, such as attributes space [19,29,47], semantic space [5,20,28,41], synthesizing features [54], using objects to create common space [51].

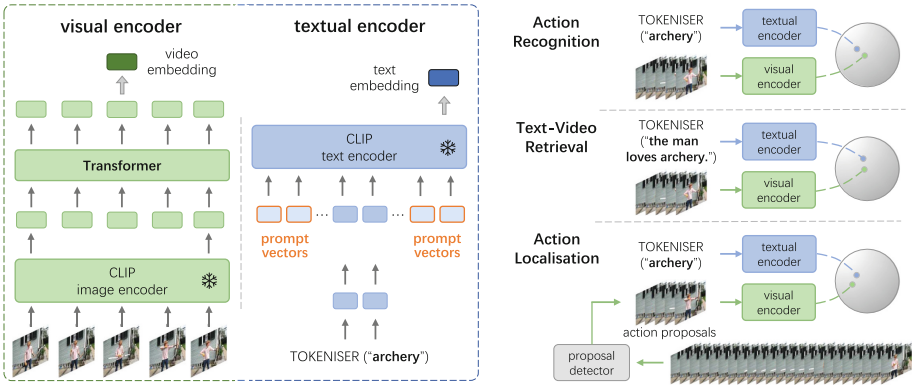

Fig. 1. Framework Overview. We prepend/append several learnable prompt vectors to the CLIP text encoder to generate action classifiers or embeddings; adopt a lightweight Transformer on top of the CLIP image encoder for temporal modeling. During training, both the image and text encoders are kept *frozen*. By optimising task-specific prompt vectors and temporal Transformer, we efficiently adapt CLIP to various video understanding tasks: action recognition, retrieval, and localisation, across closed-set, few-shot, and zero-shot scenarios.

Video Action Localisation aims to detect and classify actions in untrimmed long videos. In general, there are two popular detection paradigms: the two-stage paradigm [11,34,44,46,65,69,73,77,88] first localises class-agnostic action proposals, which covers correct segments with high recall, then classifies and refines each proposal. The one-stage paradigm [7,35,45,58,82,84] combines localisation and classification, *i.e.* densely classifies each frame into actions or background.

Concurrent Work. Several recent papers [21,31,86,89,90] also explore prompt learning for efficient transfer from pre-trained CLIP to downstream image tasks. In the video domains, [50,72] propose to end-to-end finetune CLIP on individual video tasks, *e.g.* action recognition and retrieval. In contrast, we favor efficient adaptation from image to video, present the first yet simple approach on prompt learning, to establish strong and wide baselines for video understanding.

3 Method

Our goal is to efficiently steer a pre-trained **I**mage-based **V**isual-**L**anguage model (**I-VL**) to tackle novel downstream tasks, which we term as model adaptation. Here, we consider resource-hungry video understanding, *i.e.* action recognition, action localisation, and text-video retrieval. To be self-contained, in Sect. 3.1, we briefly review the pre-training and inference of I-VL models; in Sect. 3.2, we describe the proposed prompt learning and temporal modeling.

3.1 Image-Based Visual-Language Model

Pre-training. Given N (image, text) pairs in one batch, the feature embeddings for image and text are computed with two individual encoders, and a dense cosine similarity matrix is calculated between all N possible (image, text) pairs. The training objective is to jointly optimise the image and text encoders, by maximizing the similarity between N correct pairs of (image, text) associations, while minimizing the similarity for $N \times (N-1)$ incorrect pairs by a symmetrical cross-entropy over the dense matrix, *i.e.* noise contrastive learning.

Note that, both encoders contain a **tokeniser** for converting image patches or language words to vectors. In particular, the input images are divided into patches and flattened into vectors, also called "visual tokens"; while the input texts are converted into vectors ("textual tokens") by a trainable look-up table.

Inference. Once trained, the I-VL model can be deployed for image classification tasks on open vocabulary (zero-shot generalisation), with the corresponding visual classifiers being generated from the text encoder Φ_{text}, which resembles the idea of hypernetwork [24]. For example, to classify an image as cat or dog, the classifiers (c_{cat} and c_{dog}) can be generated as:

$$c_{\text{cat}} = \Phi_{\text{text}}(\text{TOKENISER}(\text{"this is a photo of } \underline{[\text{cat}]}\text{"}))$$
$$c_{\text{dog}} = \Phi_{\text{text}}(\text{TOKENISER}(\text{"this is a photo of } \underline{[\text{dog}]}\text{"}))$$

and "this is a photo of $[\cdot]$" is a handcrafted prompt template, which has shown to be effective for image classification [60].

Discussion. Despite the tremendous success on "zero-shot" image classification, the I-VL model has also shown to be sensitive to the handcrafted prompt template, clearly posing limitations on its efficient adaptation for novel downstream tasks, where the expert knowledge might be difficult to condense or unavailable. Therefore, we consider to automate such prompt design procedures, exploring efficient approaches to adapt the pre-trained image-based visual-language model for novel downstream tasks, with minimal training.

3.2 Prompting CLIP for Video Understanding

In general, we believe that prompt learning on I-VL models will shine in video tasks, for two main reasons: 1) Video tasks are resource-hungry. From the *data* perspective, video-text pairs are harder to collect than image-text pairs. From the computation perspective, given the same budget, training on image-text pairs enables the model to learn more *diversity*. Thus, it is more cost-effective to train large-scale I-VL models, and prompt them for efficient video understanding. 2) Videos are composed of frame sequences, establishing temporal dependencies on powerful image-based models is a natural and economical choice (Fig. 1).

Next, we start by formulating the problem scenario and notations; then introduce the idea for efficient model adaptation through prompt learning; lastly, we augment the I-VL image encoder via temporal modeling, disambiguating the

actions that require temporal reasoning. Among the I-VL models, CLIP [60] is the publicly available milestone, we hence base this research on it, but the same technique should be applicable to other I-VL models as well.

Problem Scenario. Given a video dataset that consists of training and validation sets, $\mathcal{D} = \{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}\}$, *e.g.* $\mathcal{D}_{\text{train}} = \{(\mathcal{V}_1, y_1), \ldots, (\mathcal{V}_n, y_n)\}$. The video $\mathcal{V}_i \in \mathbb{R}^{T\times H\times W\times 3}$ can range from seconds (recognition and retrieval), to minutes long (localisation). Respectively, y_i either refers to *one* of the $\mathcal{C}_{\text{train}}$ action labels in the text format for recognition, *e.g.* $y_i =$ 'archery'; or *dense* action category labels of T timestamps for localisation, *e.g.* $y_i \in \mathbb{R}^{T\times \mathcal{C}_{\text{train}}}$; or fine-grained text descriptions for retrieval, *e.g.* $y_i =$ 'fry the onion in a pan'.

In the closed-set scenario, the action categories for training and evaluation are the same, *i.e.* $\mathcal{C}_{\text{train}} = \mathcal{C}_{\text{val}}$; while in the zero-shot case, the action categories for training and evaluation are disjoint, *i.e.* $\mathcal{C}_{\text{train}} \cap \mathcal{C}_{\text{val}} = \emptyset$.

Model Adaptation by Learning Prompts. The goal here is to steer pre-trained CLIP to perform various video tasks with minimal training. In specific, we strive for efficient model adaptation by prepending/appending a sequence of continuous random vectors ("prompt vectors") with the textual tokens. While training, both the image and text encoders of CLIP are kept *frozen*, and the gradients will flow through the text encoder to only update the prompt vectors. Ultimately, these learnable vectors end up constructing "virtual" prompt templates that can be understood by the text encoder, and to generate desired classifiers or query embeddings, as detailed below.

(a) Action Recognition considers to classify the video clip or snippet into one of action categories. To generate the action classifier, we construct the "virtual" prompt template via feeding the tokenised category name into the pre-trained text encoder Φ_{text}, for instance:

$$c_{\text{archery}} = \Phi_{\text{text}}(a_1, \ldots, \text{TOKENISER}(\text{"archery"}), \ldots, a_k)$$
$$c_{\text{bowling}} = \Phi_{\text{text}}(a_1, \ldots, \text{TOKENISER}(\text{"bowling"}), \ldots, a_k)$$

where $a_i \in \mathbb{R}^D$ denotes the i-th prompt vector, consisting of several learnable parameters, and D is the vector dimension. c_{archery} refers to the generated classifier for the action of "archery". Note that, the prompt vectors $\{a_i\}$ are shared for all action categories, thus they are only task-specific.

(b) Action Localisation considers to localise and classify actions in untrimmed long videos. Here, we adopt the two-stage paradigm [11,87] to first detect potential class-agnostic action proposals (detailed in Sect. 4.1), and followed by performing action classification on these detected proposals.

(c) Text-Video Retrieval considers to jointly learn visual and textual embeddings that pair the video and its corresponding textual description. In contrast to action recognition, where a video snippet is coarsely labeled by an action category, the text description in video retrieval contains more fine-grained details,

usually a sentence. We here similarly TOKENISE the entire sentence, and feed the tokenised results to the text encoder with learnable prompt vectors, to generate the *query embedding* for each sentence.

(d) Summary. Generally speaking, learning prompts for model adaptation offers the following benefits: 1) As both classification and retrieval can be tackled with one framework, with classifiers or query embeddings generated from text, either category names or free-form descriptions, all tasks can utilise *one* shared backbone, yet achieve competitive performance (Sect. 4); 2) Adapting to novel tasks only requires to optimise several prompt vectors, facilitating the few-shot problem (Table 3); 3) It enables to make better use of abundant training data, and further generalise beyond the closed-set categories (Table 4 and 6).

Temporal Modeling. As for pre-training, CLIP has thoroughly relied on the (image, text) pairs, posing clear pros on cons. On the one hand, the training data can be easily crawled from the web, which enables to learn much richer contents under a given compute constraint. However, on the other hand, it ignores the temporal component of the visual scene, and struggles to recognise the dynamic events, *e.g.* push or pull, open or close. In this section, we bridge this image-to-video gap with a simple and lightweight temporal modeling module.

To be specific, we upgrade the CLIP image encoder Φ_{image} into a video one Φ_{video}, by attaching a Transformer Encoder on top of frame-wise features from the frozen image encoder:

$$v_i = \Phi_{\text{video}}(\mathcal{V}_i) = \Phi_{\text{TEMP}}(\{\Phi_{\text{image}}(I_{i1}), \ldots, \Phi_{\text{image}}(I_{iT})\})$$

where Φ_{TEMP} refers to temporal modeling module, which is a multi-layer Transformer Encoder, consisting of Multi-head Self-attention, Layer Norm, and MLPs. To indicate the temporal order, we also add learnable temporal positional encoding onto image features. $v_i \in \mathbb{R}^{T \times D}$ is dense feature embeddings of T frames.

Training Loss. Given a batch of (video, text) training pairs, the visual stream ends up with dense frame-wise feature embeddings (v_i); while for the textual stream, depending on the considered downstream tasks, it ends up with a set of action classifiers ($c_i \in \mathcal{C}_{\text{action}}$) or textual query embeddings ($c_i \in \mathcal{C}_{\text{query}}$).

For action recognition and text-video retrieval, we further compute the video-snippet-level feature by taking the mean pooling of the dense features:

$$\overline{v}_i = \Phi_{\text{POOL}}(v_i) \in \mathbb{R}^{1 \times D} \tag{1}$$

For action localisation, we take the mean pooling of the dense features within each detected action proposal, to obtain the proposal-level feature. And for simplicity, we also denote this proposal-level feature as $\overline{v}_i$.

During training, we jointly optimise the textual prompt vectors and temporal Transformer, such that the video snippet (proposal) features and its paired

Table 1. Ablation study for closed-set action recognition. Baseline-I denotes the "zero-shot" CLIP inference with the handcrafted prompt template ("a photo of [·]."). Baseline-II is the standard practice for training linear probe on the CLIP image encoder. TFM is the number of temporal Transformer layers.

Model	Prompt	Temporal	K-400			K-700		
			TOP1	TOP5	AVG	TOP1	TOP5	AVG
Baseline-I [60]	Hand-craft	✗	–	–	–	–	–	52.4
Baseline-II [60]	✗	✗	–	–	–	–	–	66.1
A0	2+X+2	✗	65.4	88.7	77.1	56.3	81.9	69.1
A1	4+X+4	✗	66.1	89.0	77.6	56.6	82.4	69.5
A2	8+X+8	✗	67.9	90.0	79.0	57.4	83.0	70.2
A3	16+X+16	✗	68.8	90.1	79.5	57.8	83.1	70.5
A4	16+X+16	1-TFM	75.8	92.9	84.4	64.2	87.3	75.8
A5	16+X+16	2-TFM	76.6	93.3	85.0	64.7	88.5	76.6
A6	16+X+16	3-TFM	76.9	93.5	85.2	64.8	88.4	76.6
A7	16+X+16	4-TFM	76.8	93.5	85.2	64.9	87.9	76.4

classifier or textual query embedding emit the highest similarity score among others. This is achieved with a simple NCE loss:

$$\mathcal{L} = -\sum_{i} \left(\log \frac{\exp(\overline{v}_i \cdot c_i/\tau)}{\sum_j \exp(\overline{v}_i \cdot c_j/\tau)} \right) \tag{2}$$

Note that, both $\overline{v}_i$ and c_j have been L2-normalised here, and τ refers to the temperature hyper-parameter for scaling. In this way, various video tasks are formulated under the same umbrella, we therefore effectively close the optimisation objective gap between CLIP pre-training and video understanding.

4 Experiments

We experiment 3 fundamental video tasks, across 10 standard datasets. In Sect. 4.2, we conduct ablation studies on action recognition. In Sect. 4.3 and 4.4, we further benchmark on action localisation and text retrieval.

4.1 Implementation Details

The image and text encoders are adopted from pre-trained CLIP (ViT-B/16). Prompt vectors and temporal Transformer are both initialized with $\mathcal{N}(0, 0.01)$. All video frames are pre-processed to 224×224 spatial resolution, the maximum number of textual tokens is 77, vector dimension $D = 512$ and τ is set to 0.07.

For action recognition, all the videos are decoded to 30 fps, and each video is sampled 16 frames with a random frame gap (gap $\in \{1, 2, 3, 4, 5, 6, 10, 15\}$) [71].

Table 2. Comparison on closed-set action recognition. On all datasets, our model performs comparably to existing methods, training far fewer parameters.

Method	HMDB-51		UCF-101		K-400		K-700	
	TOP1	TOP5	TOP1	TOP5	TOP1	TOP5	TOP1	TOP5
I3D [10]	74.3	–	95.1	–	71.6	90.0	58.7	81.7
S3D-G [76]	75.9	–	96.8	–	74.7	93.4	–	–
R(2+1)D [70]	74.5	–	96.8	–	72.0	90.0	–	–
TSM [43]	–	–	–	–	74.7	–	–	–
R3D-50 [26]	66.0	–	92.0	–	–	–	54.7	–
NL-I3D [74]	66.0	–	–	–	76.5	92.6	–	–
X3D-XXL [14]	–	–	–	–	80.4	94.6	–	–
TimeSformer-L [3]	–	–	–	–	80.7	94.7	–	–
Ours (A5)	66.4	92.1	93.6	99.0	76.6	93.3	64.7	88.5

The temporal positional encodings consist of each frame's index and the frame sampling gap (video playing speed). For video retrieval, we utilise the 16-frame input with a random frame gap (gap $\in \{10, 15, 30\}$). The model is optimised using AdamW [49] with a learning rate of 10^{-4}, and a batch size of 64 videos.

For action localisation, we follow the two-stage paradigm: class-agnostic proposal detection and proposal classification. To obtain high-quality action proposals, we first divide the entire video into equal-frame segments; then use the CLIP image encoder with one Transformer layer to extract frame-wise embeddings; finally feed these embeddings to the off-the-shelf proposal detectors [42,80]. Note that, our method is flexible to the choice of proposal detectors, and we do not innovate on such candidate proposal procedures. To generate proposal classifiers, we adopt the same implementation as for action recognition.

4.2 Action Recognition

Datasets and Metrics. HMDB-51 [37] contains 7k videos of 51 categories. Its standard split is 3570 training videos, and 1530 testing videos. **UCF-101** [67] contains 13k videos spanning 101 categories. The standard split is 9537 training videos and 3783 testing videos. **Kinetics-400** [36] (K-400) covers around 230k 10-second video clips sourced from YouTube. **Kinetics-700** [9] (K-700) is simply an extension of K-400, with around 650k video clips. For evaluation metrics, we report the standard TOP1 and TOP5 accuracy, and the average of the two.

Closed-set Action Recognition is the common scenario, where the model is trained and evaluated on videos from the same categories, *i.e.* $\mathcal{C}_{\text{train}} = \mathcal{C}_{\text{val}}$. For comprehensive comparisons, we here adopt the standard splits of four datasets.
• *Ablation Studies* are conducted on two largest benchmarks. Table 1 presents the results for prompt learning and temporal modeling. The prompt here follows the

Table 3. Comparison on few-shot action recognition. Baseline-I refers to the "zero-shot" CLIP inference with handcrafted prompts. $\mathcal{C}_{\text{ALL}}$ refers to the case where the method is evaluated on all action categories of the corresponding dataset, rather than only 5-way classification, *e.g.* 400 categories for K-400.

Method	K-shot	N-way	Prompt	Temporal	UCF-101	HMDB-51	K-400
CMN [91]	5	5	–	–	–	–	78.9
TARN [4]	5	5	–	–	–	–	78.5
ARN [85]	5	5	–	–	83.1	60.6	82.4
TRX [59]	5	5	–	–	96.1	75.6	85.9
Baseline-I [60]	–	5	Hand-craft	✗	91.9	68.9	95.1
Ours	5	5	✓	✗	**98.3**	**85.3**	**96.4**
	5	5	✓	✓	97.8	84.9	96.0
Baseline-I [60]	–	$\mathcal{C}_{\text{ALL}}$	Hand-craft	✗	64.7	40.1	54.2
Ours	5	$\mathcal{C}_{\text{ALL}}$	✓	✗	77.6	56.0	57.1
	5	$\mathcal{C}_{\text{ALL}}$	✓	✓	**79.5**	**56.6**	**58.5**

format of $[a_1, .., a_k, \text{X}, a_{k+1}, .., a_{2k}]$. Note that, although we prepend and append the equal number of prompt vectors, the optimisation can perfectly learn to ignore any of these vectors, thus, we do not ablate other prompt formats.

As the baselines, we compare with the official results reported in the original CLIP [60]. Specifically, Baseline-I refers to the "zero-shot" inference with handcrafted prompt templates ("a photo of [·]."), and Baseline-II denotes the standard practice for training linear classifiers on top of the pre-trained CLIP image encoder with the considered downstream datasets.

Generally speaking, training more text prompt vectors brings consistent improvements on both TOP1 and TOP5 accuracy; In addition, adding temporal modeling also brings immediate benefits, with average gains of 4.9% and 5.3% on K-400 and K-700. However, it gives diminishing returns as more Transformer layers are added. Overall, all the results suggest that, both the prompt learning and temporal modeling are essential. While comparing with Baseline-I, the A3 model demonstrates a performance boost of 18.1%, clearly showing the benefits of learning prompt vectors over handcrafted ones. Moreover, even with fewer trainable parameters (only 16K), the A3 model also surpasses Baseline-II, with 4.4% gains, showing the superiority of prompting adaptation.

For all the following action recognition experiments, we inherit the best practice from the ablation studies, *i.e.* prepend/append 16 prompt vectors to category names, and only use two Transformer layers (5M parameters) for temporal modeling, for its best trade-off on performance and computational cost.

• *Comparison to SOTA*. Table 2 compares our method with existing state-of-the-art approaches on four popular action recognition benchmarks. Overall, on all datasets, our model performs comparably with the competitors, although we only need to train *far fewer* parameters (around 5M), *i.e.* two Transformer layers and several prompt vectors, advocating efficient model adaptation.

Table 4. Ablation study for zero-shot action recognition on K-700. The model is trained on 400 action categories and evaluated on the other 300 disjoint categories. Baseline-I refers to the results from the CLIP zero-shot evaluation.

Model	Prompt	Temporal	TOP1	TOP5	AVG
Baseline-I [60]	Hand-craft	✗	52.4	77.3	64.9
B0	4+X+4	✗	57.4	83.3	70.4
B1	8+X+8	✗	57.7	82.6	70.2
B2	16+X+16	✗	**58.4**	82.6	70.5
B3	32+X+32	✗	57.5	**84.6**	**71.1**
B4	16+X+16	1-TFM	47.9	76.8	62.4
B5	16+X+16	2-TFM	45.5	75.4	60.5
B6	16+X+16	3-TFM	45.6	75.2	60.4

Few-Shot Action Recognition aims to classify videos with only a few training samples, in this section, we benchmark on two different settings. The first part follows the previous literature [8,59,85], that is, evaluates the standard K-shot, N-way classification; while the second one considers a more challenging setting that classifies all categories with K-shot support samples. As baselines, in both settings, we use the "zero-shot" CLIP inference with handcrafted templates.

• 5-*Shot*-5-*Way Setting*. For fair comparisons, this setting adopts the publicly accessible few-shot splits. Specifically, for HMDB-51 and UCF-101, we follow [85] to collect 10 and 21 testing action categories respectively; while for K-400, we follow [59,91] to collect 24 testing categories, each containing 100 videos. During training, we sample 5 categories (ways) from the above data, with 5 videos (shots) from each category, and utilise the remaining data for evaluation. To ensure the statistical significance, we conduct 200 trials with random samplings.

Table 3 presents the average TOP1 accuracy for three datasets. Our method (with/without temporal modeling) outperforms all previous methods by a large margin, around 10% on HMDB-51 and K-400, showing the superiority.

• 5-*Shot*-$\mathcal{C}$-*Way Setting*. Here, we further consider a more challenging scenario: scaling the problem up to classifying all action categories in the dataset with only 5 training samples per category, for example, $\mathcal{C}_{\text{ALL}} = 400$ for K-400, $\mathcal{C}_{\text{ALL}} = 101$ for UCF-101. Specifically, on each dataset, we sample 5 videos (shots) from the training set for each category, to form the few-shot support set, and then measure performance on the corresponding standard testing set.

For this experiment setting, we conduct 10 random sampling rounds, and also record the average TOP1 accuracy in Table 3. Comparing to the 5-way classification, the $\mathcal{C}$-way setting is clearly more challenging, our model (with/without temporal modeling) still shows promising results. While comparing to the Baseline-I, our performance gains on both UCF-101 and HMDB-51 are around 15%.

Zero-Shot Action Recognition refers to the novel scenario, where videos for training and validation are from different action categories, *i.e.* $\mathcal{C}_{\text{train}} \cap \mathcal{C}_{\text{val}} = \emptyset$.

Table 5. Comparison on closed-set action localisation. Baseline-III adopts the same first-stage proposal detector as our method, but uses the original CLIP with handcrafted prompts as the second-stage proposal classifier. AVG is the average mAP in [0.3:0.1:0.7] on THUMOS14, and [0.5:0.05:0.95] on ActivityNet1.3.

Method	Date	Modality	THUMOS14						ActivityNet1.3			
			0.3	0.4	0.5	0.6	0.7	AVG	0.5	0.75	0.95	AVG
CDC [64]	2017	RGB+Flow	40.1	29.4	23.3	13.1	7.9	22.8	45.3	26.0	0.2	23.8
TALNET [11]	2018	RGB+Flow	53.2	48.5	42.8	33.8	20.8	39.8	38.2	18.3	1.3	20.2
DBS [23]	2019	RGB+Flow	50.6	43.1	34.3	24.4	14.7	33.4	–	–	–	–
A2NET [80]	2020	RGB+Flow	58.6	54.1	45.5	32.5	17.2	41.6	43.6	28.7	3.7	27.8
GTAD [79]	2020	RGB+Flow	66.4	60.4	51.6	37.6	22.9	47.8	50.4	34.6	9.0	34.1
BSN++ [68]	2021	RGB+Flow	59.9	49.5	41.3	31.9	22.8	41.1	51.3	**35.7**	8.3	**34.9**
AFSD [42]	2021	RGB+Flow	**67.3**	**62.4**	**55.5**	**43.7**	**31.1**	**52.0**	**52.4**	35.3	6.5	34.4
TALNET [11]	2018	RGB	42.6	–	31.9	–	14.2	–	–	–	–	–
A2NET [80]	2020	RGB	45.0	40.5	31.3	19.9	10.0	29.3	39.6	25.7	2.8	24.8
Baseline-III	2022	RGB	36.3	31.9	25.4	17.8	10.4	24.3	28.2	18.3	3.7	18.2
Ours	2022	RGB	**50.8**	**44.1**	**35.8**	**25.7**	**15.7**	**34.5**	**44.0**	**27.0**	**5.1**	**27.3**

Specifically, we split K-700 into two parts, with $\mathcal{C}_{\text{train}} = 400$ categories for training, and the remaining $\mathcal{C}_{\text{val}} = 300$ categories for evaluation.

As a baseline, we evaluate the CLIP with handcrafted prompt templates. As reported in Table 4, our model achieves 6.0% gains on TOP1 accuracy over the Baseline-I, showing the effectiveness of prompt learning for zero-shot recognition. Interestingly, the number of learnable prompt vectors does not make a difference, and adding temporal modeling diminishes the performance gain. We conjecture this is because the additional Transformer layer could specialise on the training set, thus harming the generalisation towards unseen action categories.

Conclusion and Discussion. Among all recognition benchmarks, we have demonstrated the effectiveness of prompt learning and temporal modeling. For closed-set recognition, even without temporal modeling, learnable prompts clearly surpass the handcrafted ones, and linear probe settings. While comparing to state-of-the-art methods, despite training *far fewer* parameters, our model still shows competitive performance. For few-shot recognition, model adaptation by prompt learning really shines, outperforming all previous methods significantly. Lastly, for zero-shot recognition, textual prompts enable to make better use of abundant training data, and further improve the generalisation beyond the seen categories.

4.3 Action Localisation

Datasets and Metrics. THUMOS14 [32] covers 413 untrimmed sports videos from 20 action categories, with an average of 15 instances per video. The standard split is 200 training videos and 213 validation videos. **ActivityNet1.3** [27] has around 20k untrimmed videos of 200 action categories. The standard split is

Table 6. Results of zero-shot action localisation. Baseline-III uses the same proposal detector as our method, but adopts the original CLIP with handcrafted prompts as the proposal classifier. Our model is trained on 75% (or 50%) action categories and tested on the remaining 25% (or 50%) action categories.

Method	Train *v.s* Test	THUMOS14						ActivityNet1.3			
		0.3	0.4	0.5	0.6	0.7	AVG	0.5	0.75	0.95	AVG
Baseline-III	75% *v.s* 25%	33.0	25.5	18.3	11.6	5.7	18.8	35.6	20.4	2.1	20.2
Ours	75% *v.s* 25%	**39.7**	**31.6**	**23.0**	**14.9**	**7.5**	**23.3**	**37.6**	**22.9**	**3.8**	**23.1**
Baseline-III	50% *v.s* 50%	27.2	21.3	15.3	9.7	4.8	15.7	28.0	16.4	1.2	16.0
Ours	50% *v.s* 50%	**37.2**	**29.6**	**21.6**	**14.0**	**7.2**	**21.9**	**32.0**	**19.3**	**2.9**	**19.6**

10,024 training videos and 4,926 validation videos. We evaluate with the mean Average Precision (mAP) at various IoU thresholds. For THUMOS14, the IoU set is $[0.3 : 0.1 : 0.7]$; as for ActivityNet1.3, the IoU set is $[0.5 : 0.05 : 0.95]$.

Closed-Set Action Localisation is the commonly adopted setting, where the model is trained and tested on videos of the same categories, *i.e.* $\mathcal{C}_{\text{train}} = \mathcal{C}_{\text{val}}$. For fair comparisons, we use the standard dataset splits as in the literature.

Table 5 reports the results. As a baseline, we adopt the same first-stage proposal detector, but utilise the original CLIP with handcrafted prompts ("this is an action of $[\cdot]$") for the second-stage proposal classifier. On both datasets, our model significantly outperforms the Baseline-III. While comparing to other existing methods that use pre-trained RGB stream, our method also demonstrates superior performance, with around 5.2% and 2.5% gains on average mAP.

Zero-Shot Action Localisation refers to the novel scenario, where the action categories for training and testing are disjoint. As we are not aware of any existing benchmarks on this challenging scenario, we initiate two evaluation settings: one is to train with 75% categories and test on the left 25% categories; the other is to train with 50% categories and test on the left 50% categories. To ensure statistical significance, we conduct 10 random samplings for data splits.

Table 6 shows the average performance. As proposals are class-agnostic, the key of two-stage localisation is the proposal classifier. For comparisons, we also implement the baseline, which uses the same proposal detector as our model, but classifies action proposals using original CLIP with handcrafted prompts. In both settings, our model shows superior performance than the Baseline-III. However, when comparing with closed-set, the zero-shot performance drops dramatically. Note that, such drop now comes from two sources: one is the recall drop from the first-stage class-agnostic proposals, and the other comes from the second-stage classification errors. See the supplementary materials for complete ablations.

Table 7. Results of text-video retrieval. E2E denotes if the model has been trained end-to-end. Baseline-IV denotes the original CLIP model with text query naïvely encoded, *i.e.* without using any prompt. As these methods are pre-trained on different datasets with variable sizes, it is unlikely to make fair comparisons.

Method	E2E	MSRVTT(9K)		LSMDC		DiDeMo		SMIT	
		R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CE [48]	✗	21.7	51.8	12.4	28.5	16.1	41.1	–	–
MMT [18]	✗	24.6	54.0	13.2	29.2	–	–	–	–
TT-CE+ [12]	✗	29.6	61.6	**17.2**	**36.5**	21.6	48.6	–	–
Baseline-IV	✗	31.2	53.7	11.3	22.7	28.8	54.6	39.3	62.8
Ours	✗	**36.7**	**64.6**	13.4	29.5	**36.1**	**64.8**	**66.6**	**87.8**
Frozen [2]	✓	31.0	59.5	15.0	30.8	34.6	65.0	–	–
CLIP4Clip [50]	✓	44.5	71.4	22.6	41.0	43.4	70.2	–	–

4.4 Text-Video Retrieval

Datasets and Metrics. MSRVTT [78] contains 10,000 videos and 200,000 captions. We train on "Training-9K" split [18], and test on "test 1k-A" [83] of 1,000 clip-text pairs. **LSMDC** [61] contains 118,081 videos of 2 to 30 seconds. We train on 7,408 validation videos, and evaluate on another 1,000 videos. **DiDeMo** [1] contains 10,464 videos annotated with 40,543 sentences. **SMIT** [55] contains more than 500k videos randomly chosen from M-MiT training set [56], and 10k validation videos. We evaluate with the average recall at K (R@K).

Results. Table 7 presents text-retrieval results on four benchmarks. Note that, we here only employ 8 learnable prompt vectors, *i.e.* [4+X+4]. This is because the pre-trained CLIP text encoder takes limited number of textual tokens up to 77, whereas the retrieval query can be long. For the cases where the tokenised text query is longer than the maximum supported length of CLIP, we simply truncate the sequence to fit our specified pattern, as such cases are few in practice.

While comparing with the Baseline-IV that denotes the results from the original CLIP with naïvely-encoded text queries, our proposed prompt learning and temporal modeling demonstrate clear benefits on all three benchmarks. Comparing with the existing approaches that are specifically targeting on text-video retrieval, our proposed method can still perform competitively, although it only requires to optimise several prompt vectors, along with two Transformer layers. Note that, as these methods are usually pre-trained on different datasets with variable sizes, this is by no means to make fair comparisons.

5 Conclusion

Building on CLIP, this paper constructs wide and strong baselines for efficient video understanding, with the simple idea of learning lightweight prompt vectors

and temporal Transformer. We evaluate on 10 popular benchmarks from: action recognition, action localisation, and text-video retrieval. Thorough comparisons and ablations are conducted to analyse the critical components. In the closed-set scenario, despite training only a few free parameters, we achieve competitive performance to the modern state-of-the-art methods. In few-shot and zero-shot scenarios, we significantly outperform existing methods on 7 public benchmarks.

Acknowledgement. This work is supported by the National Key Research and Development Program of China (No. 2020YFB1406801), 111 plan (No. BP0719010), STCSM (No. 18DZ2270700), State Key Laboratory of UHD Video and Audio Production and Presentation, the UK EPSRC Programme Grant Visual AI (EP/T028572/1), and a Google-DeepMind Scholarship.

References

1. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the International Conference on Computer Vision (2017)
2. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: a joint video and image encoder for end-to-end retrieval. Proceedings of the International Conference on Computer Vision (2021)
3. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: Proceedings of the International Conference on Machine Learning (2021)
4. Bishay, M., Zoumpourlis, G., Patras, I.: TARN: temporal attentive relation network for few-shot and zero-shot action recognition. In: Proceedings of the British Machine Vision Conference (2019)
5. Brattoli, B., Tighe, J., Zhdanov, F., Perona, P., Chalupka, K.: Rethinking zero-shot video classification: end-to-end training for realistic applications. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
6. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems (2020)
7. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.C.: End-to-end, single-stream temporal action detection in untrimmed videos. In: Proceedings of the British Machine Vision Conference (2019)
8. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
9. Carreira, J., Noland, E., Hillier, C., Zisserman, A.: A short note on the kinetics-700 human action dataset. arXiv preprint arXiv:1907.06987 (2019)
10. Carreira, J., Zisserman, A.: Quo Vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
11. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster R-CNN architecture for temporal action localisation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)

12. Croitoru, I., et al.: TeachText: crossmodal generalized distillation for text-video retrieval. In: Proceedings of the International Conference on Computer Vision (2021)
13. Dwivedi, S.K., Gupta, V., Mitra, R., Ahmed, S., Jain, A.: ProtoGAN: towards few shot learning for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
14. Feichtenhofer, C.: X3D: expanding architectures for efficient video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
15. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast networks for video recognition. In: Proceedings of the International Conference on Computer Vision (2019)
16. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
17. Frome, A., et al.: Devise: a deep visual-semantic embedding model. In: Advances in Neural Information Processing Systems (2013)
18. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12349, pp. 214–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58548-8_13
19. Gan, C., Yang, T., Gongi, B.: Learning attributes equals multi-source domain generalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
20. Gan, C., Yang, Y., Zhu, L., Zhao, D., Zhuang, Y.: Recognizing an action using its name: a knowledge-based approach. Int. J. Comput. Vision **120**, 61–77 (2016)
21. Gao, P., et al.: Clip-adapter: better vision-language models with feature adapters. arXiv preprint arXiv:2110.04544 (2021)
22. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. In: Association for Computational Linguistics (2021)
23. Gao, Z., Wang, L., Zhang, Q., Niu, Z., Zheng, N., Hua, G.: Video imprint segmentation for temporal action detection in untrimmed videos. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
24. Ha, D., Dai, A., Le, Q.: Hypernetworks. In: Proceedings of the International Conference on Learning Representations (2016)
25. Han, T., Xie, W., Zisserman, A.: Temporal alignment network for long-term video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
26. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d CNNs retrace the history of 2d CNNs and ImageNet? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
27. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: ActivityNet: a large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
28. Jain, M., Van Gemert, J.C., Mensink, T., Snoek, C.G.: Objects2action: classifying and localizing actions without any video example. In: Proceedings of the International Conference on Computer Vision (2015)
29. Jain, M., Van Gemert, J.C., Snoek, C.G.: What do 15,000 object categories tell us about classifying and localizing actions? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)

30. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: Proceedings of the International Conference on Machine Learning (2021)
31. Jia, M., et al.: Visual prompt tuning. arXiv preprint arXiv:2203.12119 (2022)
32. Jiang, Y.G., et al.: THUMOS challenge: action recognition with a large number of classes (2014). https://crcv.ucf.edu/THUMOS14/
33. Jiang, Z., Xu, F.F., Araki, J., Neubig, G.: How can we know what language models know? Trans. Assoc. Comput. Linguist. **8**, 423–438 (2020)
34. Ju, C., Zhao, P., Chen, S., Zhang, Y., Wang, Y., Tian, Q.: Divide and conquer for single-frame temporal action localization. In: Proceedings of the International Conference on Computer Vision (2021)
35. Ju, C., Zhao, P., Chen, S., Zhang, Y., Zhang, X., Tian, Q.: Adaptive mutual supervision for weakly-supervised temporal action localization. arXiv preprint arXiv:2104.02357 (2021)
36. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
37. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: Proceedings of the International Conference on Computer Vision (2011)
38. Lei, J., et al.: Less is more: ClipBERT for video-and-language learning via sparse sampling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
39. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2021)
40. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. In: Association for Computational Linguistics (2021)
41. Li, Y., hung Hu, S., Li, B.: Recognizing unseen actions in a domain-adapted embedding space. In: IEEE International Conference on Image Processing (2016)
42. Lin, C., et al.: Learning salient boundary feature for anchor-free temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
43. Lin, J., Gan, C., Han, S.: TSM: temporal shift module for efficient video understanding. In: Proceedings of the International Conference on Computer Vision (2019)
44. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: BMN: boundary-matching network for temporal action proposal generation. In: Proceedings of the International Conference on Computer Vision (2019)
45. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the ACM International Conference on Multimedia (2017)
46. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: boundary sensitive network for temporal action proposal generation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208, pp. 3–21. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_1
47. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2011)
48. Liu, Y., Albanie, S., Nagrani, A., Zisserman, A.: Use what you have: video retrieval using representations from collaborative experts. In: Proceedings of the British Machine Vision Conference (2019)

49. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations (2019)
50. Luo, H., et al.: CLIP4Clip: an empirical study of clip for end to end video clip retrieval. arXiv preprint arXiv:2104.08860 (2021)
51. Mettes, P., Thong, W., Snoek, C.G.M.: Object priors for classifying and localizing unseen actions. Int. J. Comput. Vision **129**, 1954–1971 (2021)
52. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
53. Miech, A., Laptev, I., Sivic, J.: Learning a text-video embedding from incomplete and heterogeneous data. arXiv preprint arXiv:1804.02516 (2018)
54. Mishra, A., Pandey, A., Murthy, H.A.: Zero-shot learning for action recognition using synthesized features. Neurocomputing **390**, 117–130 (2020)
55. Monfort, M., et al.: Spoken moments: learning joint audio-visual representations from video descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
56. Monfort, M., et al.: Multi-moments in time: Learning and interpreting models for multi-action video understanding. IEEE Trans. Pattern Anal. Mach. Intell., 1 (2021)
57. Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: First International Workshop on Multimedia Intelligent Storage and Retrieval Management (ACM Multimedia Conference) (1999)
58. Nawhal, M., Mori, G.: Activity graph transformer for temporal action localization. arXiv preprint arXiv:2101.08540 (2021)
59. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal relational cross transformers for few-shot action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
60. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning (2021)
61. Rohrbach, A., et al.: Movie description. Int. J. Comput. Vision **123**, 94–120 (2017)
62. Schick, T., Schütze, H.: Exploiting cloze questions for few shot text classification and natural language inference. In: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (2021)
63. Shin, T., Razeghi, Y., IV, R.L.L., Wallace, E., Singh, S.: AutoPrompt: eliciting knowledge from language models with automatically generated prompts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2020)
64. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
65. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
66. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems (2014)
67. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)

68. Su, H., Gan, W., Wu, W., Qiao, Y., Yan, J.: BSN++: complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)
69. Tan, J., Tang, J., Wang, L., Wu, G.: Relaxed transformer decoders for direct action proposal generation. In: Proceedings of the International Conference on Computer Vision (2021)
70. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
71. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
72. Wang, M., Xing, J., Liu, Y.: ActionCLIP: a new paradigm for video action recognition. arXiv preprint arXiv:2109.08472 (2021)
73. Wang, Q., Zhang, Y., Zheng, Y., Pan, P.: RCL: recurrent continuous localization for temporal action detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
74. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
75. Weston, J., Bengio, S., Usunier, N.: WSABIE: scaling up to large vocabulary image annotation. In: Proceedings of the International Joint Conference on Artificial Intelligence (2011)
76. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219, pp. 318–335. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_19
77. Xu, H., Das, A., Saenko, K.: R-C3D: region convolutional 3d network for temporal activity detection. In: Proceedings of the International Conference on Computer Vision (2017)
78. Xu, J., Mei, T., Yao, T., Rui, Y.: MSR-VTT: a large video description dataset for bridging video and language. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
79. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-TAD: sub-graph localization for temporal action detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
80. Yang, L., Peng, H., Zhang, D., Fu, J., Han, J.: Revisiting anchor mechanisms for temporal action localization. IEEE Trans. Image Process. **29**, 8535–8548 (2020)
81. Yao, L., et al.: FILIP: fine-grained interactive language-image pre-training. In: Proceedings of the International Conference on Learning Representations (2022)
82. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
83. Yu, Y., Kim, J., Kim, G.: A joint sequence fusion model for video question answering and retrieval. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 487–503. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_29
84. Zhang, C., Wu, J., Li, Y.: ActionFormer: localizing moments of actions with transformers. arXiv preprint arXiv:2202.07925 (2022)

85. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P.H.S., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12350, pp. 525–542. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_31
86. Zhang, R., et al.: Tip-adapter: training-free clip-adapter for better vision-language modeling. arXiv preprint arXiv:2111.03930 (2021)
87. Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., Tian, Q.: Bottom-up temporal action localization with mutual regularization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12353, pp. 539–555. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58598-3_32
88. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: Proceedings of the International Conference on Computer Vision (2017)
89. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134 (2021)
90. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
91. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 782–797. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_46
92. Zhu, L., Yang, Y.: Label independent memory for semi-supervised few-shot video classification. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 273–285 (2020)
93. Zhu, X., Toisoul, A., Perez-Rua, J.M., Zhang, L., Martinez, B., Xiang, T.: Few-shot action recognition with prototype-centered attentive learning. In: Proceedings of the British Machine Vision Conference (2021)