Vita-CLIP: Video and text adaptive CLIP via Multimodal Prompting

Syed Talal Wasim[1] Muzammal Naseer[1] Salman Khan[1,2]
Fahad Shahbaz Khan[1,3] Mubarak Shah[4]
[1]Mohamed bin Zayed University of AI [2]Australian National University
[3]Linköping University [4]University of Central Florida

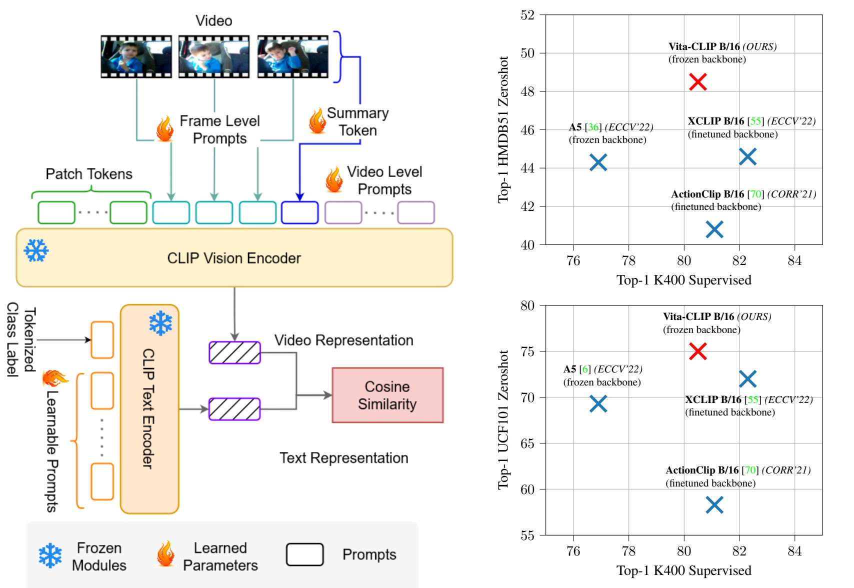
Abstract

Adopting contrastive image-text pretrained models like CLIP towards video classification has gained attention due to its cost-effectiveness and competitive performance. However, recent works in this area face a trade-off. Finetuning the pretrained model to achieve strong supervised performance results in low zero-shot generalization. Similarly, freezing the backbone to retain zero-shot capability causes significant drop in supervised accuracy. Because of this, recent works in literature typically train separate models for supervised and zero-shot action recognition. In this work, we propose a multimodal prompt learning scheme that works to balance the supervised and zero-shot performance under a single unified *training. Our prompting approach on the vision side caters for three aspects: 1) Global* video-level *prompts to model the data distribution; 2) Local* frame-level *prompts to provide per-frame discriminative conditioning; and 3) a* summary prompt *to extract a condensed video representation. Additionally, we define a prompting scheme on the text side to augment the textual context. Through this prompting scheme, we can achieve state-of-the-art zero-shot performance on Kinetics-600, HMDB51 and UCF101 while remaining competitive in the supervised setting. By keeping the pretrained backbone frozen, we optimize a much lower number of parameters and retain the existing general representation which helps achieve the strong zero-shot performance. Our codes/models will be released at https://github.com/TalalWasim/Vita-CLIP.*

1. Introduction

In the image classification domain, multimodal image-text pretrained models such as CLIP [58], ALIGN [31] and Florence [75] have shown the capability of learning generalized representations. These models, trained on large-scale language-image pairs in a contrastive manner, have remarkable zero-shot capabilities and transfer well to a variety of downstream tasks. However, training a similar model for the task of video recognition is not feasible both in terms of gathering large-scale video-text pairs, which can suffer from alignment problems [30], and is also exponentially more computationally expensive due to multiple frames being processed per video. Therefore, there has been a recent push in the research community to effectively adopt the pretrained image-text models for the task of video recognition, while maintaining their zero-shot capabilities. In this regard, existing methods can be divided into two categories. Some take inspiration from recent prompt learning methods [25, 32, 77, 81, 82] and propose a prompt learning scheme either on the text [36] or vision [55, 70] side, along with additional transformer layers for improved temporal learning. Others prefer an end-to-end CLIP finetuning scheme for video tasks [51, 55, 70]. However, the problem with these methods is that they either fail to effectively leverage learning on both the text and vision sides [36, 55] or end up losing the zero-shot generalization of CLIP by finetuning the vision decoder [47] or the backbone [51, 55, 70]. In summary, the existing approaches can steer the model *either* towards good zero-shot generalization *or* better supervised learning on video tasks. Since real-world tasks require both supervised and zero-shot capabilities, our work investigates the following question: *Can we develop a* unified *model for videos that performs well for both supervised learning and zero-shot generalization tasks?*

In pursuit of the aforementioned question, we propose a multimodal prompting-based Video and text adaptive CLIP. To effectively adapt the pretrained image-text CLIP model to videos, we consider two important aspects. Firstly, one needs to preserve the generalization capabilities of the original pretrained CLIP backbone and secondly, it must be able to effectively adapt to the video domain. In this regard, we propose to keep the entire backbone frozen and learn additional lightweight modules to adapt the model for videos. On this point, for the vision side, we aim to explicitly exploit the temporal information in videos which is lacking in the frozen image model. Our approach models video information at three levels: first via *global video-level prompts*



(a) Proposed Prompting Scheme

(b) Zero-shot accuracy (HMDB51, UCF101) vs supervised accuracy (Kinetics-400).

Figure 1. An **overview of the proposed prompting scheme** (*left*) alongside the trade-off which we attempt to balance between supervised and zero-shot performance (*right*). **(a)** Our prompting approach adds learnable parameters to learn visual and temporal information in videos at three levels: a *summary prompt* to learn a condensed representation of the video, *video-level prompts* to model global distribution shifts needed to adapt to video domain and *frame-level prompts* to enrich local discriminative information in each frame. On the text side, we learn prompts to adapt the language representations for videos. **(b)** The trade-off plots showing zero-shot vs. supervised performance comparison for ours and recent CLIP-based video approaches. Note that existing SoTA [55] trains two separate models for zero-shot and supervised settings while our method offers a unified model with the same training for both settings.

that learn the overall distribution characteristics of video data *e.g.*, motion and dynamics; secondly, inspired by [53], *local frame-level prompts* which model per frame discriminative information by directly conditioning on classification tokens of all frames; and thirdly by a *summary prompt* that distills the entire video sequence response in a single concise summary vector.

Additionally, to better model the textual context we propose to use a learnable context on the text encoder. The reason why this is particularly important is that the textual information is quite limited in the available video datasets. Instead of having per-sample text descriptions, we are limited to using class labels as text descriptions. Inspired by [82], we propose a prompt learning method on the text side to better model the textual context and to augment the video class label descriptions. An overview of our method with the trade-off it seeks to balance is presented in Fig. 1. The main contributions of this work are as follows:

- We propose a multimodal prompting approach Vita-CLIP for videos that learns video and text-specific context vectors to efficiently adapt the image-text pretrained CLIP model to video recognition tasks.
- On the vision side, we explicitly model the temporal information and the video data distribution. Our prompt learning method aggregates the discriminative information from each frame in a clip with every other frame, while also providing per-layer learning capacity to better capture the data distribution. On the language side, our approach learns complimentary semantic context to better adapt the language representations.
- We evaluate our approach on supervised as well as generalization tasks and demonstrate a sound balance between both aspects using a *single* unified model. Specifically, on zero-shot tasks, we obtain 4.0%, 3.0% and 2.2% gains over the recent SoTA X-CLIP [55] on HMDB-51, UCF-101, and Kinetics-600 datasets respectively.

2. Related work

Vision-Language (VL) Models: VL models [31,58,75] consists of an image and a text encoder and are trained on large-scale image-text pairs in a contrastive manner to learn a common feature space between images and textual labels. The semantic supervision driven by text allows models like CLIP [58] to learn fine-grained visual concepts which are transferable to many downstream tasks; semantic segmentation [27, 60, 80], object detection [18], point cloud classification [78], and video classification [74]. Importantly, these models allow 'zero-shot' knowledge transfer. In the video domain, there exist some models trained with video-text pairs for applications such as video retrieval [3,41,52]. However, these models are not trained on large amounts of video-text data. In this work, we propose a novel approach to induce temporal cues within the pretrained VL model, CLIP, to enhance its 'zero-shot' generalization on videos.

Video Recognition: The conventional techniques for spatiotemporal learning for video recognition progressed from hand-crafted features [16, 38, 68] to end-to-end deep learning methods [40]. Among neural network-based approaches, 3D convolutional networks (CNNs) [11, 15, 22, 65] learn spatiotemporal representation directly from RGB video data, while other methods deploy dedicated 2D CNNs [23, 34, 69, 72] and learn spatial and dynamic information within separate networks before fusing them together. The trade-off between 2D/3D networks for videos has been explored in [67, 73, 83]. Recently, Transformer [17] based architectures have emerged for video recognition [4, 7, 19, 54, 59]. In this work, we propose to adopt a pretrained multimodal Transformer [58] for spatiotemporal learning.

Prompt Learning: Prompting was proposed in NLP domain [35, 48] and it refers to generating task-specific instructions to get the desired behavior from language models. These instructions can be created manually [9] or learned by training discrete [26, 35, 61, 63] or continuous vectors [42,44]. Prompt learning has recently been explored in vision problems to transfer knowledge from large-scale models to downstream tasks. The current prompting techniques are applied to both uni-models *e.g*., ViTs trained on images [17] as well as multimodal models such as CLIP. For the case of ViTs, [5,33] train learnable prompts to steer pretrained vision transformers [17, 49]. On the other hand, methods like [66,81,82] introduce learnable vectors into the text encoder of CLIP for transfer learning to image recognition tasks. In contrast, we propose to learn multimodal video prompts to steer both vision and text encoders of CLIP simultaneously for spatiotemporal learning on videos.

Adapting VL Models for Videos: CLIP model has been fully fine-tuned on video-based retrieval and recognition tasks [51, 70]. Ju *et al*. [36] transfer the zero-shot generalization capability of CLIP to videos by learning prompts on the text encoder inputs and two transformer layers on the frame-level visual representations from the image encoder to model temporal context. However, directly using the CLIP image encoder for videos leads to a lack of temporal information within earlier blocks of the CLIP vision encoder and as a consequence, such an approach shows less generalization than full-fine tuning [70]. Similarly, [55] proposes a cross-frame attention module to model long-range inter-frame dependencies in videos and uses text prompt generation conditioned on video and text representations for better generalization. In contrast to these methods, we introduce a learnable video prompting module within the image and text encoder of CLIP to model temporal cues without full fine-tuning and demonstrate a good trade-off between generalization and fully supervised performance.

3. Vita-CLIP: Methodology

Our approach, Vita-CLIP, works to adapt pretrained image-based vision-language models for videos using a multimodal prompting scheme that aims to retain both the strong generalization capability (zero-shot performance) as well as good supervised performance. Vita-CLIP allows utilizing the existing image-language pretrained model rather than training one from scratch for videos.

This section presents our approach. We start with an overview of the vision/text encoders in Sec. 3.1, followed by a detailed explanation of our multimodal prompt learning scheme in Sec. 3.2. This is further divided into vision (Sec. 3.2.1) and text-side prompt learning (Sec. 3.2.2). Finally, we outline our learning objective in Sec. 3.3.

3.1. Video and Text Encoding

As stated earlier, we wish to adopt the pretrained image-text models to videos in a manner that we retain both the pretrained generalized representation, while also achieving competitive fully-supervised performance with methods that employ finetuning on the text and/or vision encoders. In that regard, we propose a multimodal vision and text prompt learning scheme that keeps both the vanilla CLIP image and text encoders frozen and introduces extra learnable parameters to adopt them for videos. From a broader perspective, we obtain video ($\mathbf{v}$) and text ($\mathbf{c}$) representations from the video (f_{θ_v}) and text (f_{θ_c}) encoders respectively. This section formally defines how these representations are obtained, while specific details on the proposed prompt learning scheme are presented in Sec. 3.2.

Video Encoder: Consider a video $V \in \mathbb{R}^{T\times H\times W\times 3}$ of spatial size $H\times W$ with T sampled frames. Each frame $t \in \{1\dots T\}$ is divided into N non-overlapping square patches of size $P\times P$ as required by the ViT architecture [17], with the total number of patches being $N = H\times W/P^2$. For each frame, all patches of shape $P\times P\times 3$ are flattened into a set of vectors and represented as $\{\mathbf{x}_{t,i} \in \mathbb{R}^{3P^2}\}_{i=1}^{N}$, where t is the frame number and i the patch number. The vectors are then projected to form token embeddings using a linear projection layer $\mathbf{P}_{emb} \in \mathbb{R}^{3P^2\times D}$, with an output dimension D for each token. An additional classification token, $\mathbf{x}_{cls} \in \mathbb{R}^{D}$, is prepended to the sequence of embedded tokens for each frame. The final per-frame token sequence fed into the video encoder is given by:

$$\mathbf{z}_t^{(0)} = [\mathbf{x}_{cls}, \mathbf{P}_{emb}^T\,\mathbf{x}_{t,1}, \cdots, \mathbf{P}_{emb}^T\,\mathbf{x}_{t,N}] + \mathbf{e}, \quad (1)$$

where $\mathbf{e} = \mathbf{e}^{sp} + \mathbf{e}^{tm}$. Here, $\mathbf{e}^{sp}$ and $\mathbf{e}^{tm}$ denote the spatial and temporal positional encodings, respectively.

From the L_v layered video encoder, we obtain the frame level representation at each layer l as follows:

$$\mathbf{z}_t^{(l)} = f_{\theta_v}^{(l)}(\mathbf{z}_t^{(l-1)}), \quad l \in \{1, \cdots, L_v\}, \quad (2)$$

where $f_{\theta_v}^{(l)}$ is the l-th layer of the video encoder.

Finally, to obtain the per-frame representation, the classification token $\mathbf{x}_{cls}$ is extracted from the output token sequence of the last layer $\mathbf{z}_t^{(L_v)}$, and projected to a dimension D' using a linear projection layer $\mathbf{P}_{out} \in \mathbb{R}^{D\times D'}$.

$$\mathbf{v}_t = \mathbf{P}_{out}^T\,\mathbf{z}_{t,0}^{(L_v)} \in \mathbb{R}^{D'}, \quad (3)$$

where $\mathbf{v}_t$ is the output representation of frame t and $\mathbf{z}_{t,0}^{(L_v)}$ is the classification token from the output sequence of the last

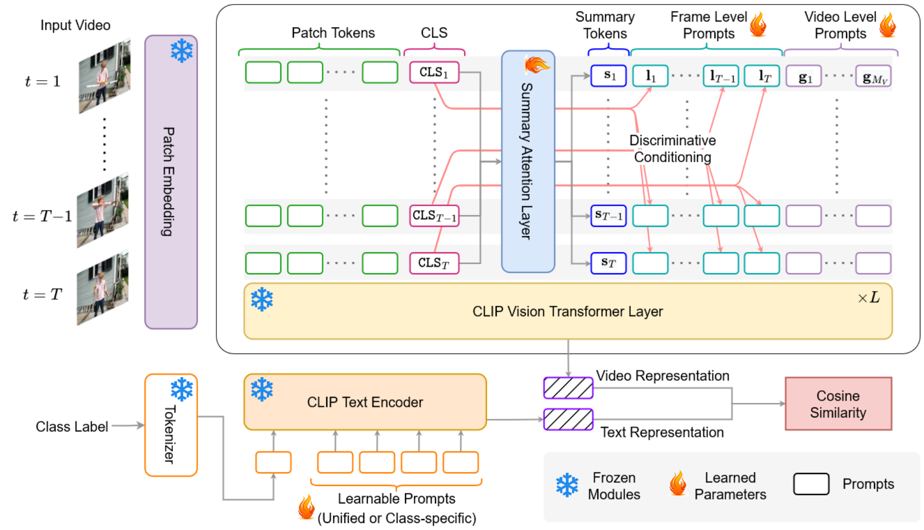


Figure 2. **Vita-CLIP Prompting Architecture:** We append multiple prompt tokens both on the vision and text encoders. On the vision encoder, we infer a Summary Token (S) which condenses the whole video token sequence which is appended with the input. Additionally, we add M_v number of Global (G) video-level prompts to learn the data distribution and (T) number of frame-level prompts conditioned on the respective frame's CLS token to reinforce discriminative information. On the text side, we add M_c number of learnable prompts to model the input context of the text encoder. Modules with (🔥) are trainable and those with (❄) are frozen.

layer of the video encoder. To obtain the video representation, the per-frame representations $\mathbf{v}_t$ are simply average-pooled to obtain the aggregate representation:

$$\mathbf{v} = \text{AvgPool}([\mathbf{v}_1, \cdots, \mathbf{v}_T]). \tag{4}$$

Text Encoder: For the input text representation, a pretrained text encoder is used with an additional text prompt learning scheme. The pretrained text encoder is a 12 layer BERT [14] model (for CLIP B/16 variant) with an embedding size of 512 and context length of 77. Each layer of the model consists of a Multi-Head Self Attention (MHSA) followed by a Feed-Forward Network (FFN). Given the text description C for a video, we use the text encoder to obtain a representation $\mathbf{c} = f_{\theta_c}(C)$. Rather than using a handcrafted prompt for the text description like "`A video of the action of {label}`", as used in recent works [70], we use a prompt learning scheme inspired by recent works on text prompting for language-image models [81, 82].

3.2. Video and text Prompt Learning

While there have been previous attempts at prompt learning to adapt language-image models to videos, they either focus on just the vision or text sides [36, 55] coupled with completely finetuning the entire vision encoder in some cases [55, 70]. To adapt our pretrained language-image model for videos, we propose a novel multimodal prompt learning scheme that keeps the pretrained model frozen, to better retain its general representation. By preserving this representation we are able to train a *single* model that can perform well both in supervised and zero-shot settings, unlike recent works [55] that require different hyper-parameter choices to produce separate models for each setting.

In that regard, our multimodal prompting aims to align the pretrained representation towards the video tasks, ensuring that both text and vision information is utilized. More specifically, on the text side, we introduce a learnable context rather than a hand-crafted prompt to allow for the text encoder to better adapt to the new video categories. On the vision side, we propose a video prompting scheme that focuses on modeling the frame-level information and inter-frame relationships as well as providing adaptability to new video data distributions. We explain our video and text prompting in Sec. 3.2.1 and Sec. 3.2.2 respectively.

3.2.1 Video Encoder Prompt Learning

For prompting on the vision encoder we have two major objectives: 1) Exploiting the temporal information by introducing information exchange between frames, and 2) providing additional parameters to adapt the CLIP image representations towards the video dataset distribution.

In that regard, we introduce three kinds of additional tokens which are appended to the token sequence $\mathbf{z}_t^{(l)}$ from frame t at layer l. Specifically, at each layer, we introduce a single *summary token* which summarises the discriminative information across all frames, T frame level *local prompt tokens* to communicate per-frame discriminative information to the rest of the frames in the clip and M_v video-level *global prompt tokens* to provide learning capacity to adapt the model to the video dataset distribution. Detailed descriptions of these types of prompt tokens are given below.

Summary Token: The summary token is inspired by the concept of message attention proposed in [55]. It is used to summarize the discriminative information from each frame in the clip and provide it back to every frame, before applying the pretrained self-attention for that layer. More specifically the summary token $\mathbf{s}_t^{(l)}$ at the l-th layer for the t-th frame is obtained by first applying a linear projection $\mathbf{P}_{sum}$ on the classification tokens $\mathbf{z}_{t,0}^{(l-1)}$ and then applying a MHSA operation between these frame-level tokens:

$$\mathbf{Z}_{0,proj}^{(l-1)} = \mathbf{P}_{sum}^{T}\mathbf{Z}_{0}^{(l-1)},$$
$$S^{(l)} = \mathrm{MHSA}(\mathrm{LN}(\mathbf{Z}_{0,proj}^{(l-1)})) + \mathbf{Z}_{0,proj}^{(l-1)}, \quad (5)$$

where $\mathbf{Z}_0^{(l-1)} = [\mathbf{z}_{1,0}^{(l-1)}, \cdots, \mathbf{z}_{T,0}^{(l-1)}]$, $\mathbf{S}^{(l)} = [\mathbf{s}_1^{(l)}, \cdots, \mathbf{s}_T^{(l)}]$ and LN indicates layer normalization. Afterward, the respective summary token is appended to the token sequence $\mathbf{z}_t^{(l-1)}$ before applying the frozen pretrained self-attention for that layer as indicated by Eq.7.

Global Prompt Tokens: The video-level *global prompt tokens* ($\mathbf{G}^{(l)} = [\mathbf{g}_1^{(l)}, \cdots, \mathbf{g}_{M_v}^{(l)}]$) are randomly initialized learnable vectors. They are used to provide the model with additional learning capacity to learn the data distribution.

Local Prompt Tokens: The frame-level *local prompt tokens* ($\mathbf{L}^{(l)} = [\mathbf{l}_1^{(l)}, \cdots, \mathbf{l}_T^{(l)}]$) are also randomly initialized learnable vectors, equal to the number of frames, T, in the clip during training, but they are conditioned on the respective classification tokens for each frame. This conditioning of $\mathbf{L}^{(l)}$ on [CLS] token $\mathbf{z}_{t,0}^{(l-1)}$ enables a top-down discriminative information flow in frame-wise learnable tokens. Each frame-level *local prompt token* is defined as:

$$\hat{\mathbf{l}}_t^{(l)} = \mathbf{l}_t^{(l)} + \mathbf{z}_{t,0}^{(l-1)}. \quad (6)$$

Finally, the tokens $\hat{\mathbf{L}^{(l)}} = [\hat{\mathbf{l}}_1^{(l)}, \cdots, \hat{\mathbf{l}}_T^{(l)}]$ and $\mathbf{G}^{(l)} = [\mathbf{g}_1^{(l)}, \cdots, \mathbf{g}_{M_v}^{(l)}]$ are appended to each frame sequence $\mathbf{z}_t^{(l-1)}$ before applying the frozen pretrained self-attention (FSA) for that layer as indicated below,

$$[\hat{\mathbf{z}}_t^{(l)}, \mathbf{S}^{(l)}, \mathbf{G}^{(l)}, \mathbf{L}^{(l)}] = \mathrm{FSA}(\mathrm{LN}([\mathbf{z}_t^{(l-1)}, \mathbf{S}^{(l)}, \mathbf{G}^{(l)}, \mathbf{L}^{(l)}])) + [\mathbf{z}_t^{(l-1)}, \mathbf{S}^{(l)}, \mathbf{G}^{(l)}, \mathbf{L}^{(l)}], \quad (7)$$

Finally, we remove the extra appended tokens and apply the feed-forward network (FFN) only on $\hat{\mathbf{z}}_t^{(l)}$ as shown below:

$$\mathbf{z}_t^{(l)} = \mathrm{FFN}(\mathrm{LN}(\hat{\mathbf{z}}_t^{(l)})) + \hat{\mathbf{z}}_t^{(l)}. \quad (8)$$

3.2.2 Text Encoder Prompt Learning

Inspired from [36, 81, 82], we also use a prompt learning scheme on the text encoder. Rather than hand-crafting a textual input based on the class labels, we model the context words using trainable vectors. More specifically the input to the text encoder, f_{θ_c}, is a sequence of tokens of the form:

$$C = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{M_c}, \{\texttt{label}\}] \quad (9)$$

where $\mathbf{u}_i, i \in \{1, \cdots, M_c\}$ is a trainable vector of the same size as the input embeddings of the text encoder, and M_c is the number of trainable unified prompts. This token sequence is then passed to the text encoder which produces the text embedding $\mathbf{c} = f_{\theta_t}(C)$.

While two different variations are possible, Unified Context (UC) (where all classes share a single set of context vectors) and Class-Specific Context (CSC) (where an independent set of context vectors is defined for each class), we use CSC in our methodology. The prompt vectors are defined as $[\mathbf{u}_i^{n_c}], i \in \{1, \cdots, M_c\}$ and $n_c \in \{1, \cdots, N_c\}$ where N_c is the total number of classes. The effectiveness of using CSC over UC is shown through ablations in Sec.4.5.

The class-specific prompts are used in all our experiments except the zero-shot ones, where novel classes can appear. For the case of zero-shot evaluation, we simply use manual prompts with any given class name.

3.3. Learning Objective

As explained above, our architecture consists of a Vision Transformer (ViT) [17] based image encoder and a BERT [14] text encoder similar to CLIP [58]. The vision and text encoders encode the video and text descriptions respectively, which are then compared using a cosine similarity objective. More formally, given a set of videos $\mathcal{V}$ and a set of text class descriptions $\mathcal{C}$, we sample video $V \in \mathcal{V}$ and an associated text description $C \in \mathcal{C}$ which are then passed to the video (f_{θ_v}) and text (f_{θ_c}) encoders respectively. This results in the video and text representations are given as:

$$\mathbf{v} = f_{\theta_v}(V \mid \mathbf{S}^{(l)}, \mathbf{G}^{(l)}, \mathbf{L}^{(l)}), \mathbf{c} = f_{\theta_t}(C). \quad (10)$$

We then define the cosine similarity loss function $\mathcal{L}_{cos}$ between the video and text representations as below:

$$\mathcal{L}_{cos}(\mathbf{v}, \mathbf{c}) = \frac{\langle \mathbf{v}, \mathbf{c} \rangle}{\|\mathbf{v}\| \, \|\mathbf{c}\|}. \quad (11)$$

We aim to maximize $\mathcal{L}_{cos}$ for the true $\mathbf{v}$ and $\mathbf{c}$ pairs and minimize otherwise.

4. Results and Analysis

4.1. Experimental Setup and Protocols

Datasets: In the supervised setting, we train on the train set of Kinetics-400 (K400) [37] and Something-Something-V2 (SSv2) [29]). We report supervised performance against existing methods in the literature on the validation sets of K400 and SSv2. For zero-shot experiments, we train on K400 training set and evaluate on three datasets: Kinetics-600 (K600) [10], HMDB51 [39] and UCF101 [64]. For zero-shot evaluation on K600, we follow [12], using the 220 new categories outside of (K400) for evaluation. Following [55], we conduct evaluation three times, each time randomly sampling 160 categories for evaluation from the 220 categories in (K600). For zero-shot evaluation on HMDB51 and UCF101, we follow [85] and report average top-1 accuracy and standard deviation on three splits of the test set.

Hyperparameters: For all experiments we train the model for 30 epochs with a cosine decay scheduler and an initial

Table 1. Comparison with state-of-the-art on Kinetics-400 [37] Supervised Training. We compare with various initializations (Random, ImageNet 1k/21k, and CLIP-400M), specifying the number of frames, views, and FLOPs. We also mention whether the models use a frozen/fine-tuned backbone and whether the method is suitable for zero-shot evaluation.

Method	Pre-training	Finetuning	Frames	Views	Top-1	Top-5	GFLOPs	Zero-shot
Initialization: Random weights								
MViTv1-B, 64×3 *(ICCV'21)* [20]	✗	✓	64	3 × 3	81.2	95.1	455	✗
Initialization: ImageNet weights								
Uniformer-B *(ICLR'22)* [43]	IN-1k	✓	32	4 × 3	83.0	95.4	259	✗
TimeSformer *(ICML'21)* [6]	IN-21k	✓	96	1 × 3	78.0	93.7	590	✗
Mformer *(NeurIPS'21)* [56]	IN-21k	✓	16	10 × 3	79.7	94.2	370	✗
Swin-B *(CVPR'22)* [50]	IN-1k	✓	32	4 × 3	80.6	94.6	282	✗
Swin-B *(CVPR'22)* [50]	IN-21k	✓	32	4 × 3	82.7	95.5	282	✗
MViTv2-B *(CVPR'22)* [45]	✗	✓	32	5 × 1	82.9	95.7	225	✗
Initialization: Large-scale image-language weights (finetuned backbone)								
ActionCLIP-B/16 *(arXiv'21)* [70]	CLIP-400M	✓	32	10 × 3	83.8	96.2	563	✓
X-CLIP-B/16 *(ECCV'22)* [55]	CLIP-400M	✓	8	1 × 1	82.3	95.4	145	✓
X-CLIP-B/16 *(ECCV'22)* [55]	CLIP-400M	✓	8	4 × 3	83.8	96.7	145	✓
X-CLIP-B/16 *(ECCV'22)* [55]	CLIP-400M	✓	16	4 × 3	84.7	96.8	287	✓
Initialization: Large-scale image-language weights (frozen backbone)								
EVL B/16 *(ECCV'22)* [47]	CLIP-400M	✗	8	1 x 3	**82.9**	-	444	✗
A6 *(ECCV'22)* [36]	CLIP-400M	✗	16	-	76.9	93.5	-	✓
Vita-CLIP B/16 ($M_c = 8, M_v = 8$)	CLIP-400M	✗	8	1 x 1	80.5	95.9	97	✓
Vita-CLIP B/16 ($M_c = 8, M_v = 8$)	CLIP-400M	✗	8	4 x 3	81.8	96.0	97	✓
Vita-CLIP B/16 ($M_c = 8, M_v = 8$)	CLIP-400M	✗	16	4 x 3	**82.9**	**96.3**	190	✓

Table 2. Comparison with supervised methods on Something-Something-V2 [29], with a mention of their zero-shot capability.

Method	Zero-shot	Top-1
Methods with Finetuned Backbone		
TRN *(ECCV'18)* [79]	✗	48.8
SlowFast *(CVPR'20)* [21]	✗	61.7
TSM *(ICCV'19)* [46]	✗	63.4
ViViT *(ICCV'21)* [4]	✗	65.9
Swin-B *(CVPR'22)* [50]	✗	69.6
Methods with Frozen Backbone		
B2 *(ECCV'22)* [36]	✓	38.1
Vita-CLIP B/16 ($M_c = 8, M_v = 8$)	✓	**48.7**

learning rate of 8×10^{-4}. Unless stated otherwise, the number of frames during training is set to 8. For evaluation, we use a single view of 8 frames in a supervised setting. During the zero-shot evaluation, we train the model with 8 frames but evaluate with a single view of 32 frames.

4.2. Supervised Experiments

In the supervised setting, we present results on K400 and SSv2 in Tab. 1 and Tab. 2 respectively. We compare against existing methods under various initializations (random, ImageNet-1k/21k [13] and CLIP400M) and in terms of GFLOPs, training frames and evaluation views.

Comparing Vita-CLIP with the ImageNet pretrained methods, we see that our models perform better or competitively against all others while maintaining much lower GFLOP counts and keeping the entire backbone frozen. We perform better than both TimeSformer [6] and Mformer [56] while having $6\times$ and $4\times$ lower GFLOPs, respectively. We perform on par with Swin-B [50] (IN-1k) while maintaining competitive results against Swin-B (IN-21k) and MViTv2-B with 2-3$\times$ lower GFLOPs. Note that each of these models has been fully trained, while our Vita-CLIP only trains the proposed prompting scheme.

Similarly, comparing Vita-CLIP with CLIP-400M pretrained methods, we achieve 3.6% better top-1 accuracy against the A6 [36] prompting method which also uses a frozen backbone similar to ours. We also perform competitively against both X-CLIP [55] and ActionCLIP [70], both of which fine-tune the pretrained backbone while maintaining a lower GFLOP count. Compared with EVL [47], which also uses a frozen backbone, our performance is save, and we additionally hold two advantages. Firstly, we have $4.5\times$ lower GFLOPs, and secondly, we retain the zero-shot capability while EVL cannot be used for zero-shot recognition.

On SSv2, we compare supervised performance against recent methods in Tab. 2. While we are lower than cross-entropy-based methods, we surpass the best vision-text-based method B6 [36], by more than 10%. Note that the performance for vision-language models is consistently lower than cross-entropy ones. This is due to the fine-grained nature of the SSv2 class descriptions, which are more difficult to differentiate compared to, for example, K400 classes.

From the above experiments, we can see that our Vita-CLIP performs better or competitive against existing methods while maintaining the capability of zero-shot inference. This can be attributed to our prompting scheme that helps capture both the per-frame variation (through the local frame-level prompts) as well as the overall distribution of the video and the dataset (through the summary token

Table 3. Comparison for zero-shot performances on HMDB51 [39] and UCF101 [64] against state-of-the-art.

Method	HMDB-51	UCF-101
Methods with Vision Training		
ASR *(ECML'17)* [71]	21.8 ± 0.9	24.4 ± 1.0
ZSECOC *(CVPR'17)* [57]	22.6 ± 1.2	15.1 ± 1.7
UR *(CVPR'18)* [84]	24.4 ± 1.6	17.5 ± 1.6
TS-GCN *(AAAI'19)* [24]	23.2 ± 3.0	34.2 ± 3.1
E2E *(CVPR'20)* [8]	32.7	48
ER-ZSAR *(ICCV'21)* [12]	35.3 ± 4.6	51.8 ± 2.9
Methods with Vision-Language Training		
ActionCLIP *(arXiv'21)* [70]	40.8 ± 5.4	58.3 ± 3.4
A5 *(ECCV'22)* [36]	44.3 ± 2.2	69.3 ± 4.2
X-CLIP-B/16 *(ECCV'22)* [55]	44.6 ± 5.2	72.0 ± 2.3
Vita-CLIP B/16 ($M_c = 8, M_v = 8$)	**48.6 ± 0.6**	**75.0 ± 0.6**

Table 4. Comparison against state-of-the-art on Kinetics-600 [10] zero-shot performance.

Method	Top-1
Methods with Vision Training	
SJE *(ICCV'15)* [2]	22.3 ± 0.6
ESZSL *(ICML'15)* [62]	22.9 ± 1.2
DEM *(CVPR'17)* [76]	23.6 ± 0.7
GCN *(arXiv'2020)* [28]	22.3 ± 0.6
ER-ZSAR *(ICCV'2021)* [12]	42.1 ± 1.4
Methods with Vision-Language Training	
X-CLIP-B/16 *(ECCV'2022)* [55]	65.2 ± 0.4
Vita-CLIP B/16 ($M_c = 8, M_v = 8$)	**67.4 ± 0.5**

and the global video-level prompts respectively).

4.3. Zero-shot Experiments

As stated earlier, in the zero-shot experiments we train our Vita-CLIP on the K400 training set with 8 frames, then perform the zero-shot evaluation on three datasets, UCF101 [64], HMDB51 [39] and K600 [10]. Notably, we utilize the *same model and hyperparameters* as used for the supervised experiments, unlike the current SoTA method X-CLIP [55] which uses a different train setting for zero-shot evaluation.

For the zero-shot setting, we simply replace the class-specific context with a tokenized class description. Our results for zero-shot performance on UCF101, HMDB51, and K600 are presented in Tab. 3 and Tab. 4 respectively. It can be seen from Tab. 3 that we outperform the previous methods by 4% and 3% respectively on HMDB51 and UCF101. Similarly, we achieve state-of-the-art zero-shot performance on K600, surpassing the previous best by 2.2%. We attribute this strong performance to both our proposed prompting scheme, as well as the fact that we retain the pretrained general representation of the CLIP backbone.

4.4. Supervised vs. Zero-shot Trade-off

In this section, we further highlight the trade-off that we attempt to balance through our proposed method. Consider Tab. 5 where the current state-of-the-art approach X-CLIP [55] has two different sets of hyper-parameters for supervised and zero-shot settings. The authors use 8 frame sampling and train for 30 epochs in the supervised setting. While in the zero-shot setting, X-CLIP trains for 10 epochs while sampling 32 frames per clip. This results in two models which only perform well in either supervised or zero-shot settings, but not both. Instead, our Vita-CLIP, which aims at retaining the generalized representation of the backbone while adapting to videos using prompt learning, is able to achieve a balance between both settings. This allows us to use a single model, trained with sampling 8 frames per clip, for a total of 30 epochs to be used in both settings.

Table 5. Comparing performance (supervised/zero-shot) and trainable parameter trade-off between X-CLIP [55] and Vita-CLIP. (*) indicates results obtained by the official repository of [55].

Method	K400 Top 1 Supervised	HMDB51 Top 1 Zeroshot	UCF101 Top 1 Zeroshot	Trainable Parameters (M)
X-CLIP B/16 (Supervised)	82.3	41.4*	67.9*	131.5
X-CLIP B/16 (Zero-shot)	78.2*	44.6	72.0	131.5
Ours B/16	80.5	48.6	75.0	38.88

Table 6. Ablations for different types of video prompts proposed in this work: Summary Token (S), Global Prompts (G) and Local Prompts (L). Text side prompting is fixed to Class-Specific Context (CSC) with $M_c = 8$ for this ablation.

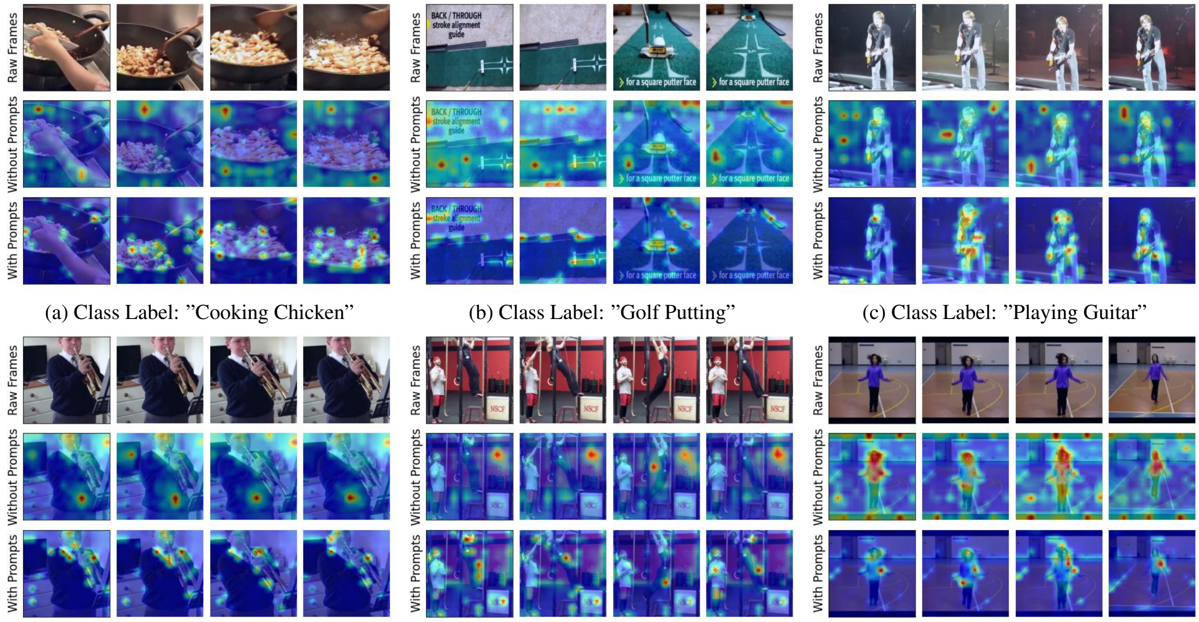
Method	Top-1
CLIP B/16 (Zero-shot)	40.10
Vita-CLIP B/16 + CSC ($M_c = 8$)	73.00
Vita-CLIP B/16 + CSC ($M_c = 8$) + G ($M_v = 8$)	77.83
Vita-CLIP B/16 + CSC ($M_c = 8$) + G ($M_v = 8$) + L	79.16
Vita-CLIP B/16 + CSC ($M_c = 8$) + G ($M_v = 8$) + L + S	80.51

4.5. Ablations

In this section, we present an ablative study on different components of our method. All experiments are performed with training on the K400 training set and testing on the validation set. All models are trained for 30 epochs, as stated earlier with 8 frames sampled per video clip.

Video Prompting: We first perform an ablation on the vision side prompting in Tab. 6. Note for this ablation, text-side prompting in all Vita-CLIP models is fixed at $M_c = 8$ using Class-Specific Context. We define a simple baseline, the zero-shot accuracy of the vanilla CLIP [58]. Building on that we add the Global video-level prompts, G ($M_v = 8$), while keeping the rest of the model frozen. This achieves 77.83% top 1 accuracy on K400. We then add the Local frame-level prompts (L) which push the model to 79.16%. The inclusion of summary token brings us up to 80.51%. This shows that the three prompting techniques are complementary and contribute to the overall accuracy of the model.

Number of Global Video-Level Prompts: We next evaluate the impact of increasing the number of Global



(a) Class Label: "Cooking Chicken" (b) Class Label: "Golf Putting" (c) Class Label: "Playing Guitar"

(d) Class Label: "Playing Trumpet" (e) Class Label: "Pull Ups" (f) Class Label: "Skipping Rope"

Figure 3. Attention Rollout [1] on sample videos showing raw frames, heatmap without our proposed prompting method, and heatmap with Vita-CLIP prompting method. For example, in actions like 'Cooking Chicken', 'Playing Guitar', 'Pull Ups', and 'Skipping Rope', our approach fixates on the important localized parts that matter the most in terms of discriminative information and motion properties.

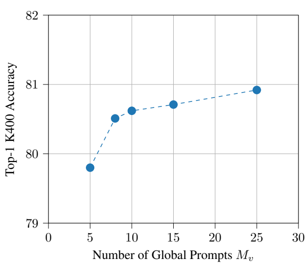


Figure 4. Ablations for number of Global video-level prompts ($\mathbf{G}^{(l)} = [\mathbf{g}_1^{(l)}, \cdots, \mathbf{g}_{M_v}^{(l)}]$) on K400 dataset. The video-side prompting includes local frame-level prompting (L) and summary token (S), while the text side prompting is fixed to Class-Specific Context (CSC) with M_c=8.

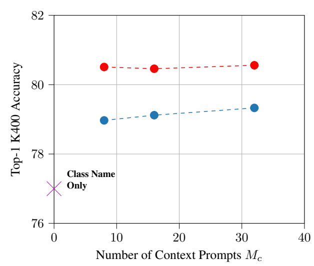


Figure 5. Ablations for the number of text prompts ($M_c = 8/16/32$) and type of text prompts: Unified Context (UC) vs. Class-Specific Context (CSC) on K400. Vision prompting is fixed to global video-level prompting G ($M_v = 8$) with local frame-level prompting L and summary token S.

video-level prompts. We test different values for the number of prompts as presented in Fig.4. We can see that the accuracy saturates around $M_v = 8$, which is why it's the default number of Global prompts we use in all experiments.

Number and Type of Text Prompts: Here, we consider the text-side prompting. We use a baseline where only the tokenized class name is used as context and evaluate two design choices: the number of text prompts M_c, and the type of text prompt, Unified Context (UC) (*i.e.* a single set of prompts for all classes), and Class-Specific Context (CSC) (*i.e.* an independent prompt set for each class). The ablation is shown in Fig. 5. It is clear that CSC gives better accuracy, which is intuitive given that there is an independent learnable context for each class. Increasing the context size beyond 8 does not give any significant gain. Thus, we chose to fix the text side prompting to CSC with $M_c = 8$.

Visualization: We illustrate the attentions of our model using the attention roll-out [1] method in Fig. 3. We compare the visualizations of our method with a baseline that does not include our proposed prompting scheme. We note that the proposed prompting scheme helps the model to focus on the salient parts and essential dynamics of the video which are relevant to the end recognition task.

5. Conclusion

We propose a multimodal prompting scheme to adopt image-language pretrained models to the task of video recognition. Existing solutions do not leverage video-text joint prompt learning and often resort to finetuning the CLIP backbone which lacks the balance between zero-shot generalization and supervised performance. Our approach strikes a balance between zero-shot and supervised performance, presenting a unified method that performs well in both settings using the same training scheme. We achieve state-of-the-art zero-shot performance on three datasets (UCF101, HMDB51, and K600) and still remain competitive with respect to supervised performance on K400 and SSv2 while training a much lower number of parameters.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online, July 2020. Association for Computational Linguistics. 8

[2] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015. 7

[3] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, pages 5803–5812, 2017. 2

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6836–6846, 2021. 3, 6

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

[6] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, pages 813–824, 2021. 2, 6

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

[8] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *CVPR*, pages 4613–4623, 2020. 7

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[10] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 5, 7

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2

[12] Shizhe Chen and Dong Huang. Elaborative rehearsal for zero-shot action recognition. In *ICCV*, pages 13638–13647, 2021. 5, 7

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 6

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4, 5

[15] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *ECCV*, pages 284–299, 2018. 2

[16] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance*, pages 65–72. IEEE, 2005. 2

[17] 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. *ICLR*, 2021. 3, 5

[18] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pages 14084–14093, 2022. 2

[19] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 3

[20] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 6

[21] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 6

[22] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019. 2

[23] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016. 2

[24] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *AAAI*, 2019. 7

[25] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 1

[26] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020. 3

[27] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021. 2

[28] Pallabi Ghosh, Nirat Saini, Larry S Davis, and Abhinav Shrivastava. All about knowledge graphs for actions. *arXiv preprint arXiv:2008.12432*, 2020. 7

[29] 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 *ICCV*, pages 5842–5850, 2017. 5, 6

[30] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *CVPR*, 2022. 1

[31] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom

Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *CoRR*, abs/2102.05918, 2021. 1, 2

[32] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 1

[33] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 3

[34] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, pages 2000–2009, 2019. 2

[35] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020. 3

[36] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*. Springer, 2022. 1, 2, 3, 4, 5, 6, 7

[37] 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, 6

[38] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, pages 275–1. British Machine Vision Association, 2008. 2

[39] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011. 5, 7

[40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2

[41] 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 *CVPR*, pages 7331–7341, 2021. 2

[42] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3

[43] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. In *ICLR*, 2022. 6

[44] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3

[45] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022. 6

[46] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 6

[47] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. *arXiv preprint arXiv:2208.03550*, 2022. 1, 6

[48] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 3

[49] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 3

[50] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022. 6

[51] 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, 3

[52] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, pages 9879–9889, 2020. 2

[53] Muzammal Naseer, Ahmad Mahmood, Salman Khan, and Fahad Khan. Boosting adversarial transferability using dynamic cues. In *ICLR*, 2023. 2

[54] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, pages 3163–3172, 2021. 3

[55] 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. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 7

[56] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *NeurIPS*, 2021. 6

[57] Jie Qin, Li Liu, Ling Shao, Fumin Shen, Bingbing Ni, Jiaxin Chen, and Yunhong Wang. Zero-shot action recognition with error-correcting output codes. In *CVPR*, pages 2833–2842, 2017. 7

[58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 5, 7

[59] Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael Ryoo. Self-supervised video transformer. In *ICCV*, June 2022. 3

[60] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, pages 18082–18091, 2022. 2

[61] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville,

and Bernt Schiele. Movie description. *IJCV*, 123(1):94–120, 2017. 3

[62] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, pages 2152–2161, 2015. 7

[63] Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020. 3

[64] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5, 7

[65] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In *WACV*, pages 625–634, 2020. 2

[66] Ximeng Sun, Ping Hu, and Kate Saenko. Dualcoop: Fast adaptation to multi-label recognition with limited annotations. *arXiv preprint arXiv:2206.09541*, 2022. 3

[67] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 3

[68] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013. 2

[69] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 2

[70] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *CoRR*, abs/2109.08472, 2021. 1, 2, 3, 4, 6, 7

[71] Qian Wang and Ke Chen. Alternative semantic representations for zero-shot human action recognition. In *ECML PKDD*, pages 87–102, 2017. 7

[72] Yunbo Wang, Mingsheng Long, Jianmin Wang, and Philip S Yu. Spatiotemporal pyramid network for video action recognition. In *CVPR*, pages 1529–1538, 2017. 2

[73] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, pages 305–321, 2018. 3

[74] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021. 2

[75] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *CoRR*, abs/2111.11432, 2021. 1, 2

[76] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, pages 2021–2030, 2017. 7

[77] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 1

[78] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *CVPR*, pages 8552–8562, 2022. 2

[79] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 6

[80] Chong Zhou, Chen Change Loy, and Bo Dai. Denseclip: Extract free dense labels from clip. *arXiv preprint arXiv:2112.01071*, 2021. 2

[81] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 1, 3, 4, 5

[82] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 1, 2, 3, 4, 5

[83] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Mict: Mixed 3d/2d convolutional tube for human action recognition. In *CVPR*, pages 449–458, 2018. 3

[84] Yi Zhu, Yang Long, Yu Guan, Shawn Newsam, and Ling Shao. Towards universal representation for unseen action recognition. In *CVPR*, pages 9436–9445, 2018. 7

[85] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, pages 8697–8710, 2018. 5