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. Fine-tuning 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...
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

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.
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 demonstrate a good performance. Specifically, [55] uses a learnable cross-frame attention module to model long-range inter-frame dependencies in videos and demonstrates a good zero-shot 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 (v) and text (c) representations from the video (f_v) and text (f_t) 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 $t \in \{1 \ldots T\}$ 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 $\{x_{t,i} \in \mathbb{R}^{P^3}\}_{i=1}^{N}$, 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 $P_{emb} \in \mathbb{R}^{P^3 \times D}$, with an output dimension $D$ for each token. An additional classification token $x_{cls} \in \mathbb{R}^{D}$, 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:

$$z_t^{(0)} = [x_{cls}, P_{emb} x_{t,1}^P, \ldots, P_{emb} x_{t,N}^P] + e,$$

where $e = e^{sp} + e^{tm}$. Here, $e^{sp}$ and $e^{tm}$ 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:

$$z_t^{(l)} = f_{\theta_v}^{(l)}(z_t^{(l-1)}), \quad l \in \{1, \ldots, L_v\},$$

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

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

$$v_t = P_{out} z_{t,0}^{(L_v)} \in \mathbb{R}^{D'},$$

where $v_t$ is the output representation of frame $t$ and $z_{t,0}^{(L_v)}$ is the classification token from the output sequence of the last
layer of the video encoder. To obtain the video representation, the per-frame representations $v_t$ are simply average-pooled to obtain the aggregate representation:

$$v = \text{AvgPool}(v_1, \ldots, v_T).$$

**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 $c = f_{\theta_t}(C)$. Rather than using a hand-crafted 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 $z^{(t)}_t$ 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 $s^{(t)}_l$ at the $l$-th layer for the $t$-th frame is obtained by first applying a linear projection $P_{sum}$ on the classification tokens $z^{(t-1)}_{t,0}$ and then applying

![Figure 2. Vita-CLIP Prompting Architecture](image)

**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 $M_v$ number of learnable prompts to model the input context of the text encoder. Modules with ($\bigcirc$) are trainable and those with ($\bigotimes$) are frozen.
a MHSA operation between these frame-level tokens:

\[
Z_{0, \text{proj}}^{(l-1)} = \mathbf{P}^T_{\text{surp}} Z_0^{(l-1)},
\]

\[
S^{(l)} = \text{MHSA}(\text{LN}(Z_{0, \text{proj}}^{(l-1)})) + Z_{0, \text{proj}}^{(l-1)},
\]

(5)

where \(Z_0^{(l-1)} = [z_0^{(l-1)}, \ldots, z_T^{(l-1)}]\), \(S^{(l)} = [s_0^{(l)}, \ldots, s_T^{(l)}]\) and LN denotes layer normalization. Afterward, the respective summary token is appended to the token sequence \(z_t^{(l-1)}\) before applying the frozen pretrained self-attention for that layer as indicated by Eq.7.

**Global Prompt Tokens:** The video-level global prompt tokens \(G^{(l)} = [g_1^{(l)}, \ldots, g_M^{(l)}]\) 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 \(L^{(l)} = [l_1^{(l)}, \ldots, l_T^{(l)}]\) 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 \(L^{(l)}\) on [CLS] token \(z_0^{(l-1)}\) enables a top-down discriminative information flow in frame-wise learnable tokens. Each frame-level local prompt token is defined as:

\[
\hat{l}_t^{(l)} = l_t^{(l)} + z_t^{(l-1)}. \tag{6}
\]

Finally, the tokens \(\hat{L}^{(l)} = [\hat{l}_0^{(l)}, \ldots, \hat{l}_T^{(l)}]\) and \(G^{(l)} = [g_1^{(l)}, \ldots, g_M^{(l)}]\) are appended to each frame sequence \(z_t^{(l-1)}\) before applying the frozen pretrained self-attention (FSA) for that layer as indicated below,

\[
[\hat{z}_t^{(l)}, S^{(l)}, G^{(l)}, L^{(l)}] = \text{FSA}(\text{LN}(\hat{z}_t^{(l-1)}, S^{(l)}, G^{(l)}, L^{(l)}))
+ [\hat{z}_t^{(l-1)}, S^{(l)}, G^{(l)}, L^{(l)}]. \tag{7}
\]

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

\[
\hat{z}_t^{(l)} = \text{FFN}(\text{LN}(\hat{z}_t^{(l)})) + \hat{z}_t^{(l)}. \tag{8}
\]

### 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, \(f_{\theta_v}\), is a sequence of tokens of the form:

\[
C = [u_1, u_2, \ldots, u_{M_c}, \{\text{label}\}] \tag{9}
\]

where \(u_i, i \in \{1, \ldots, M_c\}\) 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 \(c = f_{\theta_v}(C)\).

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 \([u_i^{nc}]\), \(i \in \{1, \ldots, M_c\}\) and \(n_c \in \{1, \ldots, N_c\}\) 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 \(f_{\theta_v}\) and text \(f_{\theta_v}\) encoders respectively. This results in the video and text representations are given as:

\[
v = f_{\theta_v}(V | S^{(l)}, G^{(l)}, L^{(l)}), c = f_{\theta_v}(C). \tag{10}
\]

We then define the cosine similarity loss function \(L_{\cos}\) between the video and text representations as below:

\[
L_{\cos}(v, c) = \frac{\langle v, c \rangle}{\|v\| \|c\|}. \tag{11}
\]

We aim to maximize \(L_{\cos}\) for the true \(v\) and \(c\) 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
while having better performance than both TimeSformer [6] and Mformer [56].

### 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, and CLIP400M), 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.

#### Table 1. Comparison with state-of-the-art on Kinetics-400 [37] Supervised Training

<table>
<thead>
<tr>
<th>Method</th>
<th>Pre-training</th>
<th>Finetuning</th>
<th>Frames</th>
<th>Views</th>
<th>Top-1</th>
<th>Top-5</th>
<th>GFLOPs</th>
<th>Zero-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVITv1-B, 64x3 (ICCV’21) [20]</td>
<td>×</td>
<td>✓</td>
<td>64</td>
<td>3 x 3</td>
<td>81.2</td>
<td>95.1</td>
<td>455</td>
<td>✓</td>
</tr>
</tbody>
</table>
| **Initialization: Random weights**
| Uniformer-B (ICLR’22) [43] | IN-1k | ✓ | 32 | 4 x 3 | 83.0 | 95.4 | 259 | × |
| TimeSformer (HCMC’21) [6] | IN-21k | ✓ | 96 | 1 x 3 | 78.0 | 93.7 | 590 | × |
| Mformer (NeurIPS’21) [56] | IN-21k | ✓ | 16 | 10 x 3 | 79.7 | 94.2 | 370 | × |
| Swin-B (CVPR’22) [50] | IN-1k | ✓ | 32 | 4 x 3 | 80.6 | 94.6 | 282 | × |
| Swin-B (CVPR’22) [50] | IN-21k | ✓ | 32 | 4 x 3 | 82.7 | 95.5 | 282 | × |
| MVITv2-B (CVPR’22) [45] | ✓ | ✓ | 32 | 5 x 1 | 82.9 | 95.7 | 225 | × |

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Zero-shot</th>
<th>Top-1</th>
</tr>
</thead>
</table>
| **Methods with Finetuned Backbone**
| TRN (ECCV’18) [79] | × | 48.8 |
| SlowFast (CVPR’20) [21] | × | 61.7 |
| TSM (ECCV’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 B16 (Mv = 8, Mf = 8) | ✓ | 48.7 |

We perform on par with Swin-B [50] (IN-1k) while maintaining competitive results against Swin-B (IN-21k) and MVITv2-B with 2-3x 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 CLIP400M 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 4.5x 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 learning rate of $8 \times 10^{-4}$. 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, 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 6x and 4x lower GFLOPs, respectively.
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 method and the pretrained general representation of the CLIP 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.

### 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 is fixed to Class-Specific Context (CSC) with $M_v = 8$ for this ablation.

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

---

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

<table>
<thead>
<tr>
<th>Method</th>
<th>HMDB-51</th>
<th>UCF-101</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR (ECML'17) [71]</td>
<td>21.8 ± 0.9</td>
<td>24.4 ± 1.0</td>
</tr>
<tr>
<td>ZSECO (CVPR'17) [57]</td>
<td>22.6 ± 1.2</td>
<td>15.1 ± 1.7</td>
</tr>
<tr>
<td>UR (CVPR'18) [84]</td>
<td>24.4 ± 1.6</td>
<td>17.5 ± 1.6</td>
</tr>
<tr>
<td>TS-GCN (AAAI'19) [24]</td>
<td>23.2 ± 3.0</td>
<td>34.2 ± 3.1</td>
</tr>
<tr>
<td>E2E (CVPR'20) [8]</td>
<td>32.7 ± 16.8</td>
<td>48.1 ± 16.8</td>
</tr>
<tr>
<td>ER-ZSAR (ICCV'21) [12]</td>
<td>33.5 ± 4.6</td>
<td>51.8 ± 2.9</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SJE (ICCV'15) [2]</td>
<td>22.3 ± 0.6</td>
</tr>
<tr>
<td>ESZSL (ICML'15) [62]</td>
<td>22.9 ± 1.2</td>
</tr>
<tr>
<td>DEM (CVPR'17) [76]</td>
<td>23.6 ± 0.7</td>
</tr>
<tr>
<td>GCN (arXiv'20) [28]</td>
<td>22.3 ± 0.6</td>
</tr>
<tr>
<td>ER-ZSAR (ICCV'20) [12]</td>
<td>42.1 ± 1.4</td>
</tr>
</tbody>
</table>

### 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].

<table>
<thead>
<tr>
<th>Method</th>
<th>K400 Top 1</th>
<th>HMDB51 Top 1</th>
<th>UCF101 Top 1</th>
<th>Trainable Parameters (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-CLIP B/16 (Supervised)</td>
<td>82.3 ± 3.0</td>
<td>41.4 ± 1.2</td>
<td>67.9 ± 1.4</td>
<td>131.5</td>
</tr>
<tr>
<td>X-CLIP B/16 (Zero-shot)</td>
<td>78.2 ± 3.0</td>
<td>44.6 ± 1.2</td>
<td>72.0 ± 1.4</td>
<td>131.5</td>
</tr>
<tr>
<td>Ours B/16</td>
<td>80.5 ± 3.0</td>
<td>48.6 ± 1.2</td>
<td>75.0 ± 1.4</td>
<td>38.88</td>
</tr>
</tbody>
</table>

### 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 $M_v = 8$ for this ablation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP B/16 (Zero-shot)</td>
<td>40.10</td>
</tr>
<tr>
<td>Vita-CLIP B/16 + CSC (M_v = 8) + G (M_c = 8)</td>
<td>73.00</td>
</tr>
<tr>
<td>Vita-CLIP B/16 + CSC (M_v = 8) + G (M_c = 8) + L</td>
<td>77.85</td>
</tr>
<tr>
<td>Vita-CLIP B/16 + CSC (M_v = 8) + G (M_c = 8) + L + S</td>
<td>79.16</td>
</tr>
<tr>
<td>Ours B/16</td>
<td>80.5 ± 3.0</td>
</tr>
</tbody>
</table>
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


[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


[36] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Manlambam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Improved multiscale vision transformers for classification and detection. In CVPR, 2022. 6


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


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