ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240499

# OmniCLIP: Adapting CLIP for Video Recognition with Spatial-Temporal Omni-Scale Feature Learning

**Mushui Liu**[a], **Bozheng Li**[a] **and Yunlong Yu**[a,*]

[a]Zhejiang University

**Abstract.** Recent Vision-Language Models (VLMs) *e.g.* CLIP have made great progress in video recognition. Despite the improvement brought by the strong visual backbone in extracting spatial features, CLIP still falls short in capturing and integrating spatial-temporal features which is essential for video recognition. In this paper, we propose OmniCLIP, a framework that adapts CLIP for video recognition by focusing on learning comprehensive features encompassing spatial, temporal, and dynamic spatial-temporal scales, which we refer to as omni-scale features. This is achieved through the design of spatial-temporal blocks that include parallel temporal adapters (PTA), enabling efficient temporal modeling. Additionally, we introduce a self-prompt generator (SPG) module to capture dynamic object spatial features. The synergy between PTA and SPG allows OmniCLIP to discern varying spatial information across frames and assess object scales over time. We have conducted extensive experiments in supervised video recognition, few-shot video recognition, and zero-shot recognition tasks. The results demonstrate the effectiveness of our method, especially with OmniCLIP achieving a top-1 accuracy of 74.30% on HMDB51 in a 16-shot setting, surpassing the recent MotionPrompt approach even with full training data. The code is available at https://github.com/XiaoBuL/OmniCLIP.

## 1 Introduction

With the surge of large-scale Internet video data, video recognition [1, 2, 17] has become increasingly critical. Recently, image-text pre-training models like CLIP [29] and ALIGN [14] have shown remarkable capabilities in the downstream image tasks [30, 46], thanks to their robust spatial feature extraction and open-vocabulary functionalities. Nevertheless, developing a similar video model demands significant computational resources. Therefore, there's a growing trend towards adapting pre-trained image-text models like CLIP for video recognition [26, 27, 40]. However, the inherent design of CLIP for static images mismatches with the dynamic nature of video, posing two primary challenges in its application for video recognition.

The first challenge is dynamic object tracking, as shown in Figure 1 (a). This requires the models not only to identify objects within each frame of the video but also to understand the actions across a series of frames over time. However, CLIP, designed for static image-text pairs, struggles with video recognition, as it poorly tracks object motion and frame continuity. The second challenge involves managing the video's continuous nature. Videos, unlike still images, evolve in time, altering object and scene characteristics. The models need to take this into account when recognizing objects, accounting for changes in their size, appearance, and behavior throughout the video, as depicted in Figure 1 (b). Thus, to address the above challenges arising from the video's dynamic nature, it is crucial to integrate temporal information across frames into CLIP to improve its capability of capturing motion trajectories and frame connections.

(a) Class: "Pulling something from behind of something"

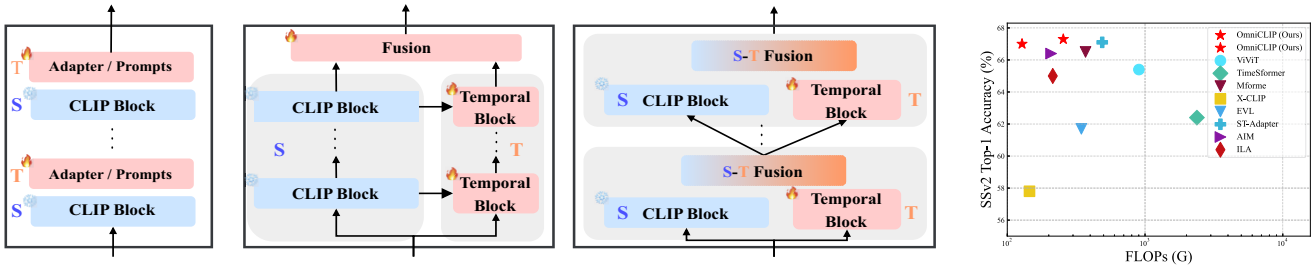
(b) Class: "playing harp"

**Figure 1.** Video recognition is challenging due to the dynamic motion and variations of objects in multi-frames.

The existing approaches for establishing temporal cues in CLIP involve integrating learnable adapters or prompts, mostly falling into two categories: temporal-embedded and temporal-side models. Temporal-embedded models, *e.g.*, ST-Adapter [27] and VitaCLIP [40], which couple space and time, involve sequential temporal-spatial integration adapters or prompts, as depicted in Figure 2 (a). While effective in handling motion and object details, they are at the cost of stacking heavy temporal-spatial modules to capture video information, constrained by lower computational efficiency. On the other hand, temporal-side models like EVL independently develop temporal modeling while maintaining static spatial representations, as demonstrated in Figure 2 (b). They offer computational efficiency but may overlook the influence of temporal dynamics on spatial details, potentially leading to suboptimal results. However, both temporal modeling manners face challenges in transferring to dynamic spatial-temporal scales, particularly in managing complex, non-linear spatial-temporal patterns, including irregular movements or the evolution of spatial structures over time.

In this paper, we argue that the integration of temporal information should be *omni-scale*, encompassing a seamless blend of spatial, temporal, and spatial-temporal integration. To achieve this, we present OmniCLIP, a novel approach that adapts CLIP for video recognition through spatial-temporal omni-scale feature learning, enabling a more sophisticated and dynamic understanding of video content, as shown in Figure 2 (c). Specifically, OmniCLIP integrates a parallel temporal adapter (PTA), tailored to enhance CLIP's capabilities in temporal modeling. PTA integrates a temporal attention mech-

* Corresponding Author. Email: yuyunlong@zju.edu.cn




(a) ST-Adapter (b) EVL (c) OmniCLIP (Ours) (d) Performance Comparison

**Figure 2.** Comparison with the recent fine-tuning CLIP to video recognition. S and T refer to spatial modeling and temporal modeling, respectively. (a) ST-Adapter. (b) EVL. (c) OmniCLIP (Ours) dynamically fuses the spatial-temporal information in parallel. (d) Performance comparison on the SSv2 dataset.

anism, meticulously tracking information across frames at consistent spatial locations. Additionally, PTA works parallel with the frozen spatial CLIP block, integrating spatial information through a straightforward learnable addition operation, thereby enabling OmniCLIP to effectively balance temporal adaptation while maintaining computational efficiency. Furthermore, OmniCLIP employs a Self-Prompt Generator (SPG) to effectively handle the dynamic interactions and irregular movements of objects across various spatial scales. Specifically, SPG leverages average pooling and a learnable projector to extract refined multi-scale information and capitalize on CLIP's strong spatial capabilities. Lastly, the combination of the PTA and SPG allows OmniCLIP to distinguish varying spatial information across frames and object scales over time, establishing a robust dynamic and spatial-temporal omni-scale framework for video recognition. This results in an efficient and competitive performance, as demonstrated in Figure 2 (d). To summarize, our highlights are as follows:

- We propose OmniCLIP, a framework to adapt CLIP for video recognition. OmniCLIP learns omni-scale video representations in spatial, temporal, and spatial-temporal scales.
- Our design incorporates a parallel temporal adapter and self-prompt generator to facilitate temporal modeling and dynamic adaptation to spatial-temporal scales. These modules enable OmniCLIP to achieve enhanced efficiency and performance in spatial-temporal processing.
- We evaluate proposed OmniCLIP with various settings *i.e.* supervised setting, few-shot setting, and zero-shot setting on different datasets *i.e.* Kinetics-400 [17], Something-to-Something v2 [10], HMDB51 [19], and UCF101 [31]. Experimental results have shown the superiority of our method, especially on the few-shot video recognition benchmark.

## 2 Related Works

**Video Recognition.** Video recognition stands as a crucial task within the video domain. The evolution of video recognition techniques has seen significant progress, transitioning from hand-crafted feature-based methods [18, 20, 34], CNN methods [11, 5, 7] to existing well-performance Transformer-based methods [1, 2, 24]. However, training video models from scratch is costly and time-consuming due to the large volume of video data. The rising of pretraining visual model [8, 29] draws great attention under such a situation and gradually grows into the priority choice of the image backbone [45, 27] for video recognition tasks.

**Vision-Language Model.** Among various pre-training models [15, 12], Vision-Language Models (VLMs) like CLIP [29] and ALIGN [14] have shown promising performance in various downstream tasks [46, 35, 23]. CLIP, for example, effectively aligns image and text representations through web-scale image-text pairs in a contrastive manner. Due to the remarkable success of the vision-language pre-training paradigm, some efforts [44, 13] have delved into utilizing video-text pairs for pre-training video-language models. OmniVL [36], for instance, introduces a unified vision-language contrastive (UniVLC) loss, aiming to maximize the exploitation of information from diverse modalities during model pre-training. While these models exhibit outstanding performance, their pre-training process frequently demands considerable resources. Consequently, methods that leverage image-based VLMs for video applications have gained popularity as they offer an efficient alternative.

**Adapting CLIP for Video Recognition** To tackle the problem of resource limitation, adapting the pre-trained image model *e.g.* CLIP for video tasks can offer an efficient and effective solution. A key aspect of this efficient adaptation process is spatial-temporal modeling. Some works [42, 43, 37, 26] fully finetune the backbone with a video-specific structure based on CLIP backbone. ActionCLIP [37] models temporal information at multiple levels including frame and video level, while X-CLIP [26] introduces a cross-frame attention mechanism for temporal information modulation. OpenVCLIP [41] is designed for tackling open-vocabulary zero-shot tasks [25], involving fine-tuning all parameters of CLIP models. Others [40, 45, 27, 22] utilize PEFT, *e.g.*, adapter structures or learnable prompts, to inject temporal learning ability into CLIP. EVL [22] utilizes a lightweight transformer decoder to capture temporal interactions among frames, and ST-Adapter [27] employs an efficient adapter with 3D convolution to concurrently learn spatial-temporal representation. Vita-CLIP [40] introduces three types of prompts to enhance temporal modeling. Prompt-based and adapter-based methods can be more resource-efficient when adapting the CLIP model for video understanding. In this work, we also transfer CLIP model with a PEFT manner to video recognition and capture the omni-scale features by a well-designed temporal adapter and dynamic spatial-temporal feature refinement. Different from TimeSformer [3] and X-CLIP [26], which also investigate temporal attention, our research focuses on the seamless integration of spatial information from the original CLIP block with temporal information derived from the temporal attention mechanism. This integration is achieved in a parallel fashion, setting our work apart from existing literature in a significant manner.

## 3 Method

### *3.1 Architecture Overview*

OmniCLIP, designed for dynamic occlusion and temporal variation in video recognition, merges spatial-temporal features for a thorough,

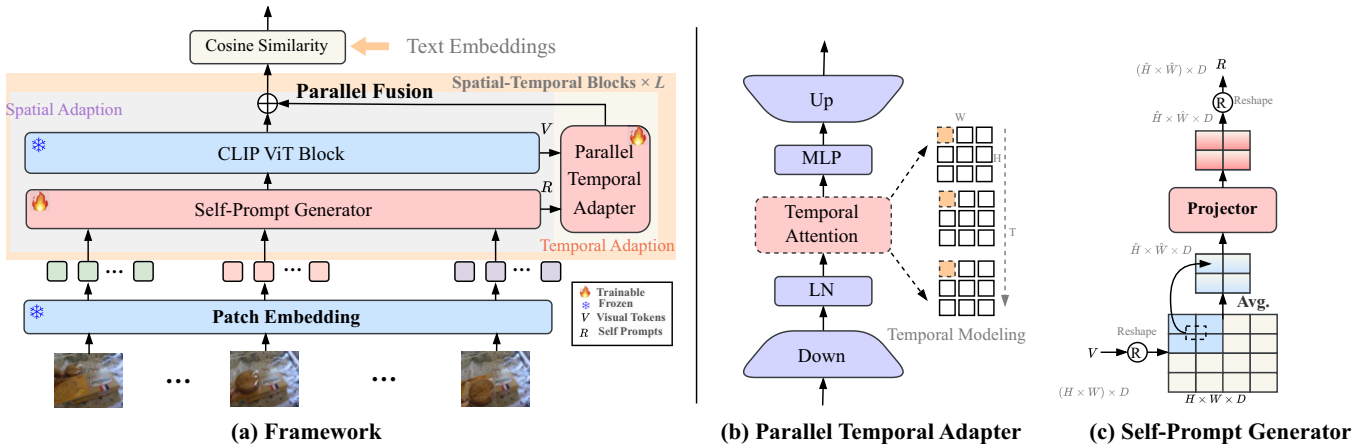



**Figure 3.** (a) The framework of our proposed OmniCLIP. (b) PTA establishes the temporal modeling. (c) SPG enhances spatial representations. The combination of PTA and SPG further improves spatial-temporal omni-scale learning.

multi-scale insight. It primarily comprises a video encoder $\theta_V$ and a text encoder $\theta_T$. The video encoder contains two main components: the parallel temporal adapter (PTA) and the self-prompt generator (SPG). Moreover, leveraging pre-trained image-text alignment, OmniCLIP employs video-specific text features to enhance zero-shot generalization and video-text alignment.

**Video Encoder** $\theta_V$ comprises $L$ spatial-temporal blocks, essential in extracting omni-scale video features, as shown in Figure 3. Each block combines a spatial ViT layer, using CLIP pre-training weights fixed during training, with a temporal adapter, actively trained for motion capture. Given a video $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$, consisting of $T$ frames, processes each frame $t$ (where $t \in \{1,\ldots,T\}$) by dividing it into $K$ non-overlapping square patches of size $P \times P$ using a ViT architecture [8], where the total patch count $K$ is $H \times W/P^2$. Each patch is initially embedded into $d$-dimensional features $X^0 = \{x_{t,j} \in \mathbb{R}^d \mid 1 \le j \le K, 1 \le t \le T\}$, where $j$ represents the patch number. Next, class tokens $e_{cls}$ are prepended to the tokens as $X^0 = \{e_{cls}, X^0\}$. The input for the spatial-temporal blocks, augmented with positional encoding $\{PE_i\}_{i=1}^N$ and temporal encoding $\{TE_i\}_{i=1}^T$, is formulated as:

$$V^0 = X^0 + PE + TE. \tag{1}$$

Subsequently, a self-prompt generator (SPG) module extracts multi-resolution video information, commencing with self-resolution prompts $R^0 = \mathrm{SPG}(V^0)$. The initial video input is subsequently augmented by concatenating it with these prompts, resulting in $V^0 = \mathrm{Concat}(V^0, R^0)$. Both the spatial ViT layer and the temporal adapter receive the identical input, $V^0$. Then, the spatial output of the $i_{th}$ layer can be derived.

$$V_s^i = \mathrm{ViT}(V^{i-1}). \tag{2}$$

In parallel, a learnable temporal adapter is used for extracting the temporal cues as:

$$V_t^i = \mathrm{PTA}(V^{i-1}). \tag{3}$$

Then a simple fusion module fuses temporal and spatial cues as:

$$V^i = V_s^i + \alpha * V_t^i, \tag{4}$$

where $\alpha$ is a learnable factor to balance the two items. Lastly, the class token $e_{cls}^{(N)}$ from the last transformer layer is projected to a latent space with a linear layer to obtain the final frame-level representation $F_{V,t}^{cls}$. Then, the aggregated video-level representation is formulated as:

$$\mathbf{F}_V = \mathrm{Avg}(\mathrm{MHA}([F_{V,1}^{cls}, F_{V,2}^{cls}, ...F_{V,t}^{cls}, ..., F_{V,T}^{cls}])), \tag{5}$$

where MHA is the multi-head attention layer and Avg denotes an average pooling operation.

Overall, the PTA module is used for temporal modeling and the SPG for spatial refinement. The combination of these two modules enables OmniCLIP to effectively extract omni-scale spatial-temporal features for video recognition.

**Text Encoder** $\theta_T$ consists of several Transformer blocks [33] and remains frozen throughout the training process. Given a video label $y$ and the class name "[CLS]", we create a description $T$ using a pre-defined template: "A video of [CLS] action". Then, we extract the text feature $\mathbf{F}_T$ using the text encoder: $\mathbf{F}_T = \theta_T(T)$. Following the approach in [26], we further enrich the text feature with video-specific prompts.

### 3.2 *Self-Prompt Generator*

Prompt learning [46] has been well explored in transferring VLMs. In our work, we introduce a unique prompt designed to enhance the representation of video spatial extraction. Motivation stems from the variation in object resolution under different perspectives within videos. Consequently, capturing information about the resolution of distinct objects is of considerable importance. To address this, we propose a self-prompt generator for augmentation. Specifically, SPG initially employs Average Pooling for the downsampling of video input $V^i$, resulting in down-sampled video features $R^i \in \mathbb{R}^{T\times \frac{K}{4}\times d}$. Subsequently, these features undergo a spatial mapping through a projector:

$$R^i = \mathrm{Projector}(\mathrm{Avg}(V^i)) \tag{6}$$

where Projector consists of two-layer MLPs. Consequently, the self-prompt generator (SPG) can enhance the extraction of spatial features in videos by learning different resolutions with the frozen ViT blocks.

### 3.3 *Parallel Temporal Adapter*

To adapt the image-based models for video recognition, previous work [27, 40, 22] usually incorporates adapters or prompts to capture the temporal information across frames. However, serial adapter structure [27] would raise the high computational costs due to the

gradient backpropagation while temporal-side fusion mechanism [22] highly relies on the representation of spatial vision backbone. Our major objectives are to construct an efficient temporal adapter that can extract temporal cues independently and enhance the representation of the origin spatial backbone with **dual-direction interaction**.

To this end, we propose to build the temporal information via a parallel temporal adapter (PTA). Specifically, PTA, which consists of a tunable self-attention layer to aggregate the same spatial location across $T$frames, is wrapped with the bottleneck structure containing the down and up projection. Given the visual tokens $V^{i-1} \in \mathbb{R}^{B\times T\times K\times D}$ in $i^{th}$ layer, where $B$ is the batch size, $T$ is the frames, $K$ is the patch numbers, and $D$ is the feature dimension, PTA firstly shift the visual tokens to $\mathbb{R}^{(BK)\times(T)\times D}$, and PTA can extract the temporal knowledge as:

$$V_t^i = \mathrm{PTA}(V^{i-1}) = \mathrm{Up}(\mathrm{Attn}(\mathrm{Down}(V^{i-1}))), \tag{7}$$

where the Down projector projects the reshaped visual tokens to a low-dimensional space for calculating the motion information and the Up projector restores the refined temporal visual feature. Note that the PTA module consists of the self-attention layers, which share the same structure as the attention block in the ViT [8] backbone while the weight is randomly initialized and tunable. By connecting the same spatial location in temporal dimensions, PTA can effectively capture the temporal cues in the video.

Note that the input of PTA contains self-prompts $R^i$, as derived by the SPG module described in eq. (6) which captures multi-resolution information. Therefore, PTA harnesses the temporal aspects of $R^i$, accumulating large-scale spatial information. This combination of SPG and PTA enables OmniCLIP to access broader spatial windows aross frames and progressively integrate spatial-temporal information.

### *3.4 Training Objectives*

Once obtained the video feature representation $\mathbf{F}_V^i$ of video sample $x_i$ and the text feature representation $\mathbf{F}_T^j$ of descriptions of class $c_j$, we calculate their cosine similarity score to quantitatively measure their semantic similarity, i.e.,

$$sim(x_i, c_j) = \frac{\langle \mathbf{F}_V^i, \mathbf{F}_T^j \rangle}{\|\mathbf{F}_V^i\| \|\mathbf{F}_T^j\|}, \tag{8}$$

where $\langle\ ,\rangle$ denotes the inner product operation. The entire model is trained by optimizing the visual-text semantic similarity when the videos and texts belong to the same class, and minimizing it when they belong to different classes. Thus, the objective function is formulated by:

$$\mathcal{L}_{obj} = \sum_i -y_i \log \mathrm{SM}\ sim(x_i, \cdot), \tag{9}$$

where SM denotes the softmax function, $y_i$ denotes the one-hot class label of $x_i$, $sim(x_i, \cdot)$ denotes the semantic similarity vector, where each element represents the semantic similarity score between $x_i$ and a class from the candidate categories.

## 4 Experiments

In this section, we present an assessment of the efficacy of our approach through its application to three distinct video recognition settings: supervised learning, few-shot learning, and zero-shot learning.

**Table 1.** Implementation details of OmniCLIP under supervised setting.

| Implementation Details | K400 | SSv2 | HMDB-51 | UCF-101 |
|---|---|---|---|---|
| **Optimization** | | | | |
| Batch size | 256 | 256 | 32 | 32 |
| Learning rate | 2e-3 | 3.5e-3 | 2e-3 | 2e-3 |
| Minimal learning rate | 2e-5 | 3.5e-5 | 2e-5 | 2e-5 |
| Training epochs | 50 | 50 | 30 | 30 |
| Optimizer | AdamW | | | |
| Learning rate schedule | Cosine | | | |
| Learning warmup epochs | 5 | | | |
| Optimizer betas | (0.9,0.98) | | | |
| **Data augmentation** | | | | |
| RandomFlip | 0.5 | | | |
| ColorJitter | 0.8 | | | |
| GrayScale | 0.2 | | | |
| Label smoothing | 0.1 | | | |
| Mixup | 0.8 | | | |
| Cutmix | 1.0 | | | |
| **Regularization** | | | | |
| Weight decay | 0.003 | 0.01 | 0.003 | 0.003 |

**Dataset Details.** We show the video datasets used in the experiments below:

- **Kinetics-400 (K400).** contains more than 230,000 10-second video clips sourced from YouTube, which have 400 categories.
- **Something-Something V2 (SSv2)** covers 174 action categories. Its standard split is 168,913 training videos, 24,777 validation videos, and 27,157 testing videos.
- **HMDB51** contains 7,000 videos and 51 categories. Its standard split is to train on 3570 videos and evaluate on another 1,530 videos.
- **UCF-101** consists of 13,000 videos spanning 101 categories. The standard split is to train on 9,537 videos and evaluate on the left 3,783 videos.

**Implementation Details.** Table 1 outlines the implementation specifics for **supervised video recognition**. For **few-shot recognition**, in accordance with [26], we maintain a batch size of 8 for both HMDB51 and UCF101 datasets, and set the number of training frames to 32. For **zero-shot recognition**, we initially train the model on the K400 dataset using 32 frames for 10 epochs, subsequently evaluating its performance on the test sets of HMDB51 and UCF101. All experiments are executed using 8 NVIDIA 24G 3090 GPUs.

### *4.1 Results of Supervised Video Recognition.*

**Datasets and Implementation Details.** We assess our method in a supervised video recognition setting across four benchmarks: Kinetics-400 (**K400**) [17], Something-Something V2 (**SSv2**) [10], HMDB51 [19], and UCF101 [31]. In all conducted experiments, the model undergoes training for a total of 50 epochs. Specifically, for the K400 and SSv2 datasets, the learning rates are set to 2e-3 and 3.5e-3, respectively. Additionally, we employ the AdamW optimizer in conjunction with a cosine annealing strategy to optimize our model. Unless otherwise stated, the input samples comprise 8 frames. For evaluation purposes, we adopt a supervised setting and utilize 4 temporal and 3 spatial views, with each view containing 8 frames.

**Results on K400.** In Table 2, we conduct a thorough comparative analysis of various competitors and our OmniCLIP on the **K400** dataset, evaluating their performance in several critical aspects. Our

**Table 2.** Comparison results of the existing competitors and OmniCLIP on **K400** dataset. The best performances are marked in **bold**. Views = #temporal clips × #spatial crops. The GFLOPs per view of each method is reported. The last column mentions that the trained model is suitable for zero-shot transfer.

| Method | Backbone | Pre-training | Frames×Views | Top-1 (%) | Top-5 (%) | GFLOPs | Zero-shot |
|---|---|---|---|---|---|---|---|
| **Methods with Vision Training** | | | | | | | |
| Uniformer-B [21] | ViT-B/16 | IN-1k | 32×4×3 | 83.0 | 95.4 | 259 | ✗ |
| TimeSformer [2] | ViT-B/16 | IN-21k | 96×1×3 | 78.0 | 93.7 | 590 | ✗ |
| Mformer [28] | ViT-B/16 | IN-21k | 16×10×3 | 79.7 | 94.2 | 370 | ✗ |
| Video-Swin [24] | Swin-B | IN-21k | 32× 4×3 | 82.7 | 95.5 | 282 | ✗ |
| **Methods with Vision-Language Training** | | | | | | | |
| ActionCLIP [37] | ViT-B/16 | CLIP-400M | 32×10×3 | 83.8 | 96.2 | 563 | ✓ |
| X-CLIP [26] | ViT-B/16 | CLIP-400M | 8×4×3 | 83.8 | 95.7 | 145 | ✓ |
| EVL [22] | ViT-B/16 | CLIP-400M | 8×1×3 | 82.9 | - | 444 | ✗ |
| ST-Adapter [27] | ViT-B/16 | CLIP-400M | 16×1×3 | 82.5 | 96.0 | 911 | ✗ |
| AIM [45] | ViT-B/16 | CLIP-400M | 8×1×3 | 83.9 | 96.3 | 606 | ✗ |
| Vita-CLIP [40] | ViT-B/16 | CLIP-400M | 16×4×3 | 82.9 | 96.3 | 190 | ✓ |
| MotionPrompt [39] | ViT-B/16 | CLIP-400M | 8×4×3 | 77.4 | 93.6 | - | ✓ |
| ILA [32] | ViT-B/16 | CLIP-400M | 8×4×3 | 84.0 | 96.6 | 149 | ✓ |
| M2-CLIP [38] | ViT-B/16 | CLIP-400M | 8×4×3 | 83.4 | 96.3 | 214 | ✓ |
| OmniCLIP (Ours) | ViT-B/16 | CLIP-400M | 8×4×3 | **84.1** | **96.7** | 130 | ✓ |

**Table 3.** Comparison results of the existing competitors and OminiCLIP on **SSv2** dataset. Views = #temporal clips × #spatial crops. The GFLOPs per view of each method is reported. The best results are marked in **bold**.

| Method | Backbone | Pre-training | Frames×Views | Top-1 (%) | GFLOPs |
|---|---|---|---|---|---|
| **Methods with Vision Training** | | | | | |
| ViViT [1] | ViT-L/14 | IN-21K+K400 | 8×1×3 | 65.4 | 903 |
| TimeSformer [2] | ViT-L/14 | IN-21K | 96×1×3 | 62.4 | 2380 |
| Mformer [28] | ViT-B/16 | IN-21K+K400 | 16×1×3 | 66.5 | 370 |
| **Methods with Vision-Language Training** | | | | | |
| X-CLIP [26] | ViT-B/16 | CLIP-400M | 8×4×3 | 57.8 | 145 |
| EVL [22] | ViT-B/16 | CLIP-400M | 16×1×3 | 61.7 | 345 |
| ST-Adapter [27] | ViT-B/16 | CLIP-400M | 8×3×1 | 67.1 | 489 |
| AIM [45] | ViT-B/16 | CLIP-400M | 8×1×3 | 66.4 | 208 |
| ILA [32] | ViT-B/16 | CLIP-400M | 8×4×3 | 65.0 | 214 |
| M2-CLIP [38] | ViT-B/16 | CLIP-400M | 8×4×3 | 66.9 | - |
| OmniCLIP (Ours) | ViT-B/16 | CLIP-400M | 8×4×3 | 67.0 | 128 |
| OmniCLIP (Ours) | ViT-B/16 | CLIP-400M | 16×4×3 | **67.3** | 255 |

**Table 4.** Comparison of supervised video recognition on both **HMDB51** and **UCF101** datasets. † denotes the results implemented with the released codes. The best results are marked in **bold**.

| Method | HMDB-51 | | UCF-101 | |
|---|---|---|---|---|
| | Top-1 (%) | Top-5 (%) | Top-1 (%) | Top-5 (%) |
| I3D [17] | 74.30 | - | 95.10 | - |
| A5 [16] | 66.40 | 92.10 | 93.60 | 99.00 |
| Vita-CLIP [40] † | 71.18 | 94.12 | 93.71 | 99.50 |
| X-CLIP [26] † | 70.94 | 93.39 | 94.37 | 99.34 |
| MotionPrompt [39] | 72.90 | 93.20 | 96.30 | 99.30 |
| OmniCLIP (Ours) | **76.64** | **95.89** | **96.30** | **99.56** |

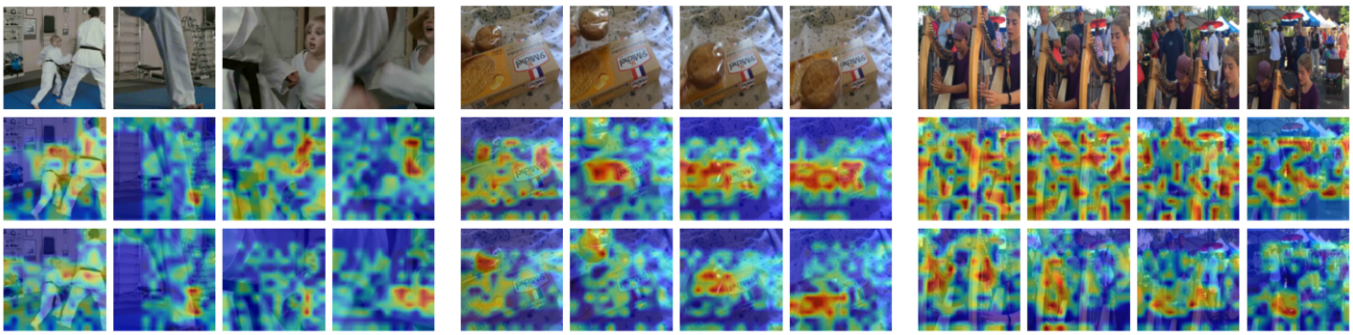
OmniCLIP exhibits the best performance, with Top-1 and Top-5 accuracy of 84.1% and 96.7%, respectively. In comparison to the most related CLIP-based competitors, including ActionCLIP [37], X-CLIP [26] Vita-CLIP [40], and M2-CLIP [38], OmniCLIP demonstrates superior performance, outperforming the second-best competitor by 0.3% and 0.4% in Top-1 and Top-5 accuracy, respectively. Notably, these improvements are particularly noteworthy given the scale of the K400 video dataset. Moreover, OmniCLIP excels in resource efficiency, achieving a minimum of 130 GFLOPs, highlighting its ability to strike a balance between accuracy and computational cost. This makes our method an ideal candidate for deployment in resource-constrained settings. Additionally, OmniCLIP showcases its adaptability and versatility through its proficiency in zero-shot transfer learning, enabling seamless transitions to other video recognition tasks.

**Results on SSv2.** SSv2 has richer temporal information and more details of action descriptions. We select recent 9 competitors for comparison. The comparison results are shown in Table 3. We train the video encoder with a fixed classifier following previous works [32]. Our OmniCLIP has demonstrated exceptional performance on the SSv2 dataset, achieving a Top-1 accuracy of 67.0% while maintaining a low computational cost of 128 GFLOPs. This impressive efficiency underscores OmniCLIP's ability to effectively capture dynamic video representations. When compared to the ST-Adapter [27], which achieves a slightly higher accuracy of 67.1% but with a significantly greater computational cost of 489 GFLOPs, OmniCLIP clearly demonstrates its superiority in balancing accuracy and computational efficiency. Furthermore, OmniCLIP exhibits remarkable adaptability, as its Top-1 accuracy improves to 67.3% when processing 16 frames on the SSv2 dataset. While this enhancement incurs a higher computational cost of 255 GFLOPs, it remains significantly lower than the ST-Adapter, highlighting OmniCLIP's flexibility under varying resource constraints. In summary, OmniCLIP offers an outstanding balance between high accuracy and low computational costs on the SSv2 dataset, making it an ideal choice for efficient video recognition tasks.

**Table 5.** Comparison results (%) under few-shot setting. We compare OmniCLIP with approaches that explicitly adapt CLIP for video recognition on **HMDB51** and **UCF101**. The best results are marked in **bold**.

| Method | HMDB51 | | | | UCF101 | | | |
|---|---|---|---|---|---|---|---|---|
| | K=2 | K=4 | K=8 | K=16 | K=2 | K=4 | K=8 | K=16 |
| Vanilla CLIP [29] | 41.9 | 41.9 | 41.9 | 41.9 | 63.6 | 63.6 | 63.6 | 63.6 |
| A5 [16] | 39.7 | 50.7 | 56.0 | 62.4 | 71.4 | 79.9 | 85.7 | 89.9 |
| ActionCLIP [37] | 47.5 | 57.9 | 57.3 | 59.1 | 70.6 | 71.5 | 73.0 | 91.4 |
| XCLIP [26] | 53.0 | 57.3 | 62.8 | 64.0 | 76.4 | 83.4 | 88.3 | 91.4 |
| MotionPrompt [39] | **55.3** | **58.7** | 64.0 | 64.6 | **82.4** | **85.8** | 89.1 | 91.6 |
| OmniCLIP | 54.1 | 58.3 | **67.3** | **74.4** | 76.7 | 84.5 | **91.5** | **95.1** |

**Results on HMDB51 and UCF101.** Table 4 presents the comparison results of five competitors and our OmniCLIP on both the HMDB51 and UCF101 datasets. Notably, all methods except for I3D, are based on the ViT-B/16 architecture. Our OmniCLIP consistently outperforms existing competitors across both Top-1 and Top-5 metrics on both datasets. On the HMDB51 dataset, OmniCLIP achieves a remarkable Top-1 accuracy of 76.64% and a Top-5 accuracy of 95.89%. This performance surpasses the second-best method, MotionPrompt [39], by significant margins of 3.74% and 2.69% in Top-1 and Top-5 accuracy, respectively. This underscores the superiority of our approach in capturing and representing the subtleties of human actions in this challenging dataset. Similarly, on the UCF101 dataset, OmniCLIP matches the highest reported Top-1 accuracy of 96.30% and sets a new benchmark with a Top-5 accuracy of 99.56%.



(a) Class: "punch" (b) Class: "pushing something..." (c) Class: "playing harp"

**Figure 4.** The attention map on sample videos, showing raw frames (the first row), heatmap with vanilla CLIP (the second row), and with our OmniCLIP (the last row). The actions like 'punch', 'pushing something from behind of something', and 'playing harp' are shown.

**Table 6.** Comparison (%) for zero-shot performances on both **HMDB51** and **UCF101** datasets. The best results are marked in **bold**.

| Method | HMDB51 | UCF101 |
|---|---|---|
| **Methods with Vision Training** | | |
| TS-GCN [9] | 23.2 ± 3.0 | 34.2 ± 3.1 |
| E2E [4] | 32.7 ± 0.0 | 48.0 ± 0.0 |
| ER-ZSRA [6] | 35.3 ± 4.6 | 51.8 ± 2.9 |
| **Methods with Vision-Language Training** | | |
| Vanilla CLIP [29] | 40.8 ± 0.3 | 63.2 ± 0.2 |
| ActionCLIP [37] | 40.8 ± 5.4 | 58.3 ± 3.4 |
| A5 [16] | 44.3 ± 2.2 | 69.3 ± 4.2 |
| X-CLIP B/16 [26] | 44.6 ± 5.2 | 72.0 ± 2.3 |
| Vita-CLIP B/16 [40] | 48.6 ± 0.6 | 75.0 ± 0.6 |
| MotionPrompt [39] | 50.1 ± 5.4 | **76.4 ± 2.5** |
| OmniCLIP | **51.3 ± 1.2** | 73.2 ± 1.0 |

## 4.2 Results of Few-Shot Classification.

**Datasets and Implementation Details.** Following [26], we conduct few-shot experiments on HMDB51 and UCF101 datasets. For each category, we randomly sample K instances for model training and use the test set for evaluation.

**Performance.** Table 5 showcases the comparison results of our OmniCLIP against five competitors on both the HMDB51 and UCF101 datasets. Notably, across various settings of K (the number of labeled samples per class), OmniCLIP consistently demonstrates competitive performance, particularly achieving the second-best results when K is set to 2 and 4. More importantly, as K increases, OmniCLIP's advantage becomes increasingly evident. Specifically, at K = 8 and K = 16, OmniCLIP achieves remarkable Top-1 accuracy rates of 67.3% and 74.4% on the HMDB51 dataset, surpassing the runner-up by significant margins of 3.3% and 9.8%, respectively. On the UCF101 dataset, OmniCLIP maintains its superiority, achieving outstanding accuracy rates of 91.5% and 95.1% at K = 8 and K = 16, respectively, outperforming the second-best method by 2.4% and 3.5%. It is noteworthy that, at K = 16, OmniCLIP even outperforms models that are trained with the full training data on the HMDB51 dataset (as shown in Table 4). This underscores the robustness and efficiency of our OmniCLIP approach in effectively capturing valuable temporal and spatial information, even with limited labeled data.

## 4.3 Results of Zero-Shot Classification.

**Datasets and Implementation Details.** Following [26], we firstly train our model on K400, and evaluate the zero-shot transfer capability on two datasets: HMDB51 and UCF101. We follow [47] and report average top-1 accuracy and standard deviation on three splits of the test set.

**Performance.** Table 6 presents the results of our zero-shot experiments. From the results, OmniCLIP emerges as a standout performer. On the UCF101 dataset, OmniCLIP achieves an outstanding average Top-1 accuracy of 73.2%, surpassing the vanilla CLIP model by a significant margin of 10.0%. Moreover, OmniCLIP excels on the HMDB51 dataset, achieving the best performance with a Top-1 accuracy of 51.3%. This further underscores the generalizability of our method across diverse video recognition tasks. However, it's worth noting that OmniCLIP's performance on UCF101 is slightly inferior to some other methods. We attribute this to the dataset's strong biases towards appearance and objects, which may limit the effectiveness of our approach in capturing temporal information and dependencies. Nevertheless, on datasets where temporal cues are more critical, such as HMDB51, OmniCLIP excels and demonstrates its superiority.

## 4.4 Further Analysis

To assess the impact of each component within OmniCLIP, we conduct ablation studies on the K400-tiny (a smaller split of the full K400) and HMDB51 datasets. Additionally, we provide visualizations for further insights.

**Impacts of different modules.** Table 7 presents a comparison between the effects of the Parallel Temporal Adapter (PTA) and the Self-Prompt Generator (SPG). Incorporating the PTA alone leads to a notable increase in performance, achieving 76.40% on K400-tiny and 75.23% on HMDB51. Similarly, the implementation of SPG independently results in enhanced outcomes, recording 73.20% on K400-tiny and 71.34% on HMDB51. The most significant improvement is observed when PTA and SPG are used in conjunction, culminating in 77.20% on K400-tiny and 76.64% on HMDB51. This indicates the synergistic benefit of integrating temporal, spatial, and dynamic spatial-temporal omni-scale features.

**Impacts of temporal ratio.** The temporal ratio refers to the compression ratio used by the Down projector in Equation (7), and it plays a crucial role in determining the efficiency of video recognition. Table 8 provides a comprehensive analysis of how varying these ratios within the temporal adapter affects recognition performance. Our findings indicate that a ratio of 1/4 yields optimal results, achieving the highest scores of 77.20% on K400-tiny and 76.64% on HMDB51. When the ratio deviates from this optimal value, either towards a lower value (1/8) or a higher one (1/2, 1), we observe a decrease in performance. This suggests that maintaining a balanced temporal ratio is crucial for achieving the best recognition outcomes.

**Table 7.** Impact (%) of different modules.

| PTA | SPG | K400-tiny | HMDB51 |
|---|---|---|---|
| ✗ | ✗ | 51.60 | 40.90 |
| ✓ | ✗ | 76.40 | 75.23 |
| ✗ | ✓ | 73.20 | 71.34 |
| ✓ | ✓ | 77.20 | 76.64 |

**Table 8.** Impact (%) of temporal ratio.

| Ratio | K400-tiny | HMDB51 |
|---|---|---|
| 1/8 | 76.84 | 75.34 |
| 1/4 | 77.20 | 76.64 |
| 1/2 | 77.13 | 76.34 |
| 1 | 76.93 | 75.94 |

**Table 9.** Impact (%) of self-prompt.

| Model | K400-tiny | HMDB51 |
|---|---|---|
| Avg. | 76.20 | 75.64 |
| Max. | 76.03 | 75.10 |
| Max. + Projector | 76.94 | 76.23 |
| Avg. + Projector | 77.20 | 76.64 |

**Table 10.** Impact (%) of different temporal adapter locations.

| PTA layers 1-3 | 4-6 | 7-9 | 10-12 | K400-tiny | HMDB51 |
|---|---|---|---|---|---|
| ✗ | ✗ | ✗ | ✗ | 51.60 | 40.90 |
| ✓ | ✗ | ✗ | ✗ | 75.20 | 74.35 |
| ✓ | ✓ | ✗ | ✗ | 76.41 | 75.83 |
| ✓ | ✓ | ✓ | ✗ | 77.01 | 76.34 |
| ✓ | ✓ | ✓ | ✓ | 77.20 | 76.64 |

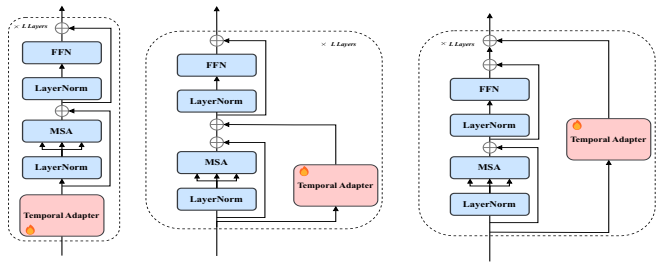
We hypothesize that overly complex temporal modules may lead to overfitting, which in turn degrades overall performance.

**Impacts of self-prompt.** Table 9 illustrates the impact of various prompt-generation methods on video recognition performance. The **Avg.** method utilizes average pooling, and the **Max.** approach applies max pooling for prompt generation. Additionally, incorporating a **Projector** represents an advanced refinement of these prompts via a learnable mechanism. The findings indicate that both average and max pooling independently deliver robust results, achieving Top-1 accuracies of 75.65% and 75.10%, respectively. This reflects their effectiveness in capturing key video content features. Moreover, the integration of the learnable projector with both **Avg.** and **Max.** further enhance their performance. We hypothesize that the **Avg.** is better suited for capturing smoother and more representative spatial features, which are crucial for video recognition, in contrast to the **Max.** Overall, the self-prompts can effectively integrate with frozen visual features, thereby adeptly capturing spatial cues.

**Impacts of different temporal adapter locations.** We explore the effects of different placements of the parallel temporal adapter within the visual branch of the ViT-B/16 model, which is divided into 12 blocks. These blocks are grouped into four sets, each containing three blocks. By strategically placing the temporal adapter in various groups, we assess the effect of how its location influences the model's performance, as detailed in Table 10. Inserting the adapter into the first group (blocks 1-3) results in a substantial improvement, with performance scores reaching 75.20% on K400-tiny and 74.35% on HMDB51. Notably, when the temporal adapter is extended across all four groups, OmniCLIP obtains its peak performance, demonstrating enhancements of 25.60% and 35.74% over the vanilla CLIP model on K400-tiny and HMDB51, respectively. This underlines the significant impact of the temporal adapter's distribution in the visual branch, showcasing a progressive enhancement in video recognition capabilities for capturing the temporal cues enhancement.

**Visualization.** Figure 4 illustrates the visual attention of the OmniCLIP. Videos are samples from various datasets for comparison and the attention map of the [CLS] token from the last layer is presented. The results demonstrate that OmniCLIP tends to focus on the moving objects (*e.g.* bread, hands) and the more important object (*e.g.* harp) of recognition, while vanilla CLIP is confused by the multi-objects and background across frames.

**Different Combinations of Temporal Blocks.** In this experiment, we assess various methods for integrating temporal adapters, including the cascade connection (Figure 5 (a)), the attention parallel connection (Figure 5 (b)), and our proposed parallel temporal adapter




(a) Cascade. (b) Attention Parallel. (c) Block Parallel (Ours).

**Figure 5.** The different combinations of the temporal adapter.

**Table 11.** Impact (%) of the combinations of the temporal adapter.

| Model | K400-tiny | HMDB51 |
|---|---|---|
| Cascade | 75.35 | 75.23 |
| Attention Parallel | 76.51 | 75.51 |
| Ours | 77.20 | 76.64 |

(Figure 5 (c)). Notably, the cascade variant of the temporal adapter can be viewed as a specialized form of the ST-Adapter, leveraging 3D convolution to capture spatial-temporal information. The results of these methodologies are summarized in Table 11. While both the cascade and attention parallel frameworks demonstrate their abilities in temporal and spatial modeling, resulting in performance improvements on the HMDB51 and UCF101 datasets, our proposed parallel temporal adapter still surpasses these strategies. Given the cascade architecture's relatively higher computational cost, we can confidently conclude that our proposed parallel temporal adapter not only outperforms other integration methods in terms of performance but also offers greater efficiency.

## 5 Conclusion

In this paper, we have presented OmniCLIP, an innovative approach that adapts CLIP model for video recognition by incorporating spatial, temporal, and dynamic spatial-temporal omni-scale features. We have designed a parallel temporal adapter (PTA) to establish effective temporal modeling, thus filling a crucial gap in the vanilla CLIP model specifically for video processing. Additionally, we have developed a self-prompt generator module (SPG) that refines spatial scale features in videos. The synergy of PTA and SPG enables OmniCLIP to effectively capture dynamic spatial-temporal features. Experimental evaluations across a range of benchmarks consistently have demonstrated that OmniCLIP excels in learning omni-scale video features, leading to notable improvements, especially in few-shot video recognition scenarios.

## Acknowledgment

This research was supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020016, the Key R&D Program of Zhejiang Province, China 2023C01043, NSFC (62002320, U19B2043, 52305590), Science and Technology Innovation 2025 Major Project of Ningbo (2023Z236).

## References


[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6836–6846, 2021.

[2] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[3] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

[4] B. Brattoli, J. Tighe, F. Zhdanov, P. Perona, and K. Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *CVPR*, pages 4613–4623, 2020.

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[6] S. Chen and D. Huang. Elaborative rehearsal for zero-shot action recognition. In *ICCV*, pages 13638–13647, 2021.

[7] R. Christoph and F. A. Pinz. Spatiotemporal residual networks for video action recognition. In *NeurIPS*, 2016.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[9] J. Gao, T. Zhang, and C. Xu. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *AAAI*, pages 8303–8311, 2019.

[10] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, pages 5842–5850, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[12] W. He, S. Fu, M. Liu, X. Wang, W. Xiao, F. Shu, Y. Wang, L. Zhang, Z. Yu, H. Li, et al. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis. *arXiv preprint arXiv:2407.07614*, 2024.

[13] J. Huang, Y. Li, J. Feng, X. Wu, X. Sun, and R. Ji. Clover: Towards a unified video-language alignment and fusion model. In *CVPR*, pages 14856–14866, 2023.

[14] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916, 2021.

[15] M. Jin, Q. Yu, H. Zhao, W. Hua, Y. Meng, Y. Zhang, M. Du, et al. The impact of reasoning step length on large language models. In *ACL*, 2024.

[16] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, pages 105–124, 2022.

[17] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[18] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, pages 275–1, 2008.

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[20] I. Laptev. On space-time interest points. *IJCV*, 64:107–123, 2005.

[21] K. Li, Y. Wang, J. Zhang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *T-PAMI*, 45(10):12581–12600, 2023.

[22] Z. Lin, S. Geng, R. Zhang, P. Gao, G. de Melo, X. Wang, J. Dai, Y. Qiao, and H. Li. Frozen clip models are efficient video learners. In *ECCV*, pages 388–404, 2022.

[23] M. Liu, W. He, Z. Lu, and Y. Yu. Sync-clip: Synthetic data make clip generalize better in data-limited scenarios. *arXiv preprint arXiv:2312.03805*, 2023.

[24] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022.

[25] Z. Lu, Z.-M. Lu, Y. Yu, Z. He, H. Luo, and Y. Zheng. Learning multiple criteria calibration for generalized zero-shot learning. *Knowledge-Based Systems*, page 112131, 2024.

[26] B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, and H. Ling. Expanding language-image pretrained models for general video recognition. In *ECCV*, pages 1–18, 2022.

[27] J. Pan, Z. Lin, X. Zhu, J. Shao, and H. Li. St-adapter: Parameter-efficient image-to-video transfer learning. In *NeurIPS*, pages 26462–26477, 2022.

[28] M. Patrick, D. Campbell, Y. Asano, I. Misra, F. Metze, C. Feichtenhofer, A. Vedaldi, and J. F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *NeurIPS*, pages 12493–12506, 2021.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[30] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, and J. Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, pages 18082–18091, 2022.

[31] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[32] S. Tu, Q. Dai, Z. Wu, Z.-Q. Cheng, H. Hu, and Y.-G. Jiang. Implicit temporal modeling with learnable alignment for video recognition. In *ICCV*, pages 19936–19947, 2023.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[34] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103:60–79, 2013.

[35] H. Wang, L. Liu, W. Zhang, J. Zhang, Z. Gan, Y. Wang, C. Wang, and H. Wang. Iterative few-shot semantic segmentation from image label text. In *IJCAI*, pages 1385–1392, 2023.

[36] J. Wang, D. Chen, Z. Wu, C. Luo, L. Zhou, Y. Zhao, Y. Xie, C. Liu, Y.-G. Jiang, and L. Yuan. Omnivl: One foundation model for image-language and video-language tasks. In *Advances in neural information processing systems*, pages 5696–5710, 2022.

[37] M. Wang, J. Xing, and Y. Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021.

[38] M. Wang, J. Xing, B. Jiang, J. Chen, J. Mei, X. Zuo, G. Dai, J. Wang, and Y. Liu. M2-clip: A multimodal, multi-task adapting framework for video action recognition. *arXiv preprint arXiv:2401.11649*, 2024.

[39] Q. Wang, J. Du, K. Yan, and S. Ding. Seeing in flowing: Adapting clip for action recognition with motion prompts learning. In *ACM MM*, pages 5339–5347, 2023.

[40] S. T. Wasim, M. Naseer, S. Khan, F. S. Khan, and M. Shah. Vita-clip: Video and text adaptive clip via multimodal prompting. In *CVPR*, pages 23034–23044, 2023.

[41] Z. Weng, X. Yang, A. Li, Z. Wu, and Y.-G. Jiang. Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In *International Conference on Machine Learning*, pages 36978–36989, 2023.

[42] W. Wu, Z. Sun, and W. Ouyang. Revisiting classifier: Transferring vision-language models for video recognition. In *AAAI*, pages 2847–2855, 2023.

[43] W. Wu, X. Wang, H. Luo, J. Wang, Y. Yang, and W. Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In *CVPR*, pages 6620–6630, 2023.

[44] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.

[45] T. Yang, Y. Zhu, Y. Xie, A. Zhang, C. Chen, and M. Li. Aim: Adapting image models for efficient video action recognition. In *ICLR*, 2023.

[46] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022.

[47] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *CVPR*, pages 8697–8710, 2018.