electronics

MDPI

Article

Adapting CLIP for Action Recognition via Dual Semantic Supervision and Temporal Prompt Reparameterization

Lujuan Deng, Jieqing Tan * and Fangmei Liu

School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450002, China; denglujuan@zzuli.edu.cn (L.D.); liufangmei@zzuli.edu.cn (F.L.)
* Correspondence: 332207050651@email.zzuli.edu.cn

Abstract: The contrastive vision–language pre-trained model CLIP, driven by large-scale open-vocabulary image–text pairs, has recently demonstrated remarkable zero-shot generalization capabilities in diverse downstream image tasks, which has made numerous models dominated by the "image pre-training followed by fine-tuning" paradigm exhibit promising results on standard video benchmarks. However, as models scale up, full fine-tuning adaptive strategy for specific tasks becomes difficult in terms of training and storage. In this work, we propose a novel method that adapts CLIP to the video domain for efficient recognition without destroying the original pre-trained parameters. Specifically, we introduce temporal prompts to realize the object of reasoning about the dynamic content of videos for pre-trained models that lack temporal cues. Then, by replacing the direct learning style of prompt vectors with a lightweight reparameterization encoder, the model can be adapted to domain-specific adjustment to learn more generalizable representations. Furthermore, we predefine a Chinese label dictionary to enhance video representation by co-supervision of Chinese and English semantics. Extensive experiments on video action recognition benchmarks show that our method achieves competitive or even better performance than most existing methods with fewer trainable parameters in both general and few-shot recognition scenarios.

Keywords: action recognition; CLIP; dual semantic supervision; temporal prompt reparameterization

Citation: Deng, L.; Tan, J.; Liu, F. Adapting CLIP for Action Recognition via Dual Semantic Supervision and Temporal Prompt Reparameterization. *Electronics* **2024**, *13*, 3348. https://doi.org/10.3390/electronics13163348

Academic Editor: George A. Tsihrintzis

Received: 29 July 2024
Revised: 20 August 2024
Accepted: 20 August 2024
Published: 22 August 2024

1. Introduction

Video understanding is an important concept in the field of computer vision, which aims to enable computers to comprehend and interpret the content presented in videos. As one of the most fundamental and challenging branches, video action recognition occupies an important position in fields such as human–computer interaction [1], intelligent transportation [2], and medical image analysis [3]. However, with the explosive growth of video volume nowadays, addressing the issue of action recognition has become urgent.

In previous research, whether it is early feature engineering [4] or the recent CNN-based architectures favored by industry professionals, e.g., 3D CNN [5], two-stream networks [6], depthwise separable convolutional networks [7], and more recently, Transformer-based models [8] that have shown outstanding performance in the visual domain, spatiotemporal representation learning has always been an enduring research topic in the video field. Although action recognition has recently achieved remarkable results by introducing Transformer to establish long-term temporal dependencies, most existing methods inevitably encounter the following problems: (i) Locating and annotating relevant human behaviors in frame-based videos is a long-term project, which is time-consuming and labor-intensive. Meanwhile, video datasets are relatively scarce and difficult to collect, especially for rare categories and videos that involve personal privacy, which weakens the model's recognition capability in these aspects. Therefore, the ability to learn "few-shot" plays a crucial driving role in enhancing the robustness of models. (ii) Follow the "closed set" learning approach, with all categories predefined and labels mapped to numbers. In other

words, the model's performance is limited when dealing with new category information that cannot be obtained during training, resulting in poor generalization. Such methods are unfriendly for many real-world applications, such as automatic video tagging, sports analysis [9], etc.

Fortunately, research on large-scale pre-training models in the field of NLP has provided a stepping stone for the computer vision community. The strong transferability and generalization presented by fine-tuning large-scale contrastive Vision-Language pre-trained Models (VLMs, e.g., CLIP [10], Florence [11]) on various downstream image tasks through the paradigm of "pre-training followed by fine-tuning" have been studied and proven [12,13]. Such significant improvements are mainly attributed to VLMs breaking away from the traditional numerical supervision style, using natural language text descriptions as supervision signals, and aligning representations to a common semantic space through contrastive loss function applied to large-scale and weakly correlated noisy image–text pairs. Therefore, given the excellent visual–language representations learned, models equipped with CLIP exhibit outstanding "few shot/zero shot" capabilities.

In the video domain, the idea of replacing image–text pairs with video–text pairs to train video–language pre-trained models [14] has been proposed. However, constructing large-scale video–text pair datasets is more difficult, and due to the influence of redundant frames in videos, the alignment between text content and the corresponding video is often permanently misaligned. These obstacles are difficult for most ordinary people to overcome. One feasible implementation is to fine-tune the pre-trained parameters of VLMs on video datasets to transfer the knowledge to the video domain. It is common to directly inflate an image-pre-trained model into a video model [5,15] and update the model parameters during training by "full fine-tuning". While the above approach performs well on specific benchmark datasets, it lacks generality and practicality; that is, if the downstream task datasets have fewer samples, the problem of over-fitting will be obvious and it is easy to damage the original good general representation of VLMs, ultimately leading to a significant discount in a model's generalization ability. Additionally, fine-tuning and saving a large number of parameters poses a dual challenge for storage and computing resources.

To overcome the above drawbacks, inspired by the research direction known as parameter-efficient transfer learning in the NLP field [16], a more economical and practical approach [17–19] has been introduced into computer vision to achieve efficient knowledge transfer from the image to the video domain. The goal is to fine-tune only a few parameters of the additional modules introduced while freezing the large pre-trained model to adapt image-level representations to video-level representations, which not only maximizes the preservation of the diversity knowledge learned by the pre-trained models but also achieves satisfactory performance. Therefore, it is crucial for researchers to design effective lightweight modules. However, most existing methods focus on transferring pre-trained image models for image tasks and pre-trained video models for video tasks [20–24]. There is relatively less exploration of the cross-domain adaptation of pre-trained image models for video tasks, mainly because image models inherently lack temporal reasoning capability.

In this work, we propose that the key to employing additional modules to reconstruct CLIP's image-level representations into video-level representations lies in the effective modeling of temporal information in vision and providing stronger semantic constraints for natural language supervision, thereby minimizing the cross-modal representation gap between vision and text. Specifically, at the visual level, inspired by the effectiveness of language model reparameterization methods [25,26], we propose a temporal prompt reparameterization encoder to replace the direct learning style of prompt vectors with implementing prompts reparameterization through a proposed encoder, aiming to make the prompts not limited by fixed parameters in the frozen CLIP visual encoder, but to establish dependency relationships with them so as to learn more generalized representations for specific domains. Finally, the reparameterized prompts are concatenated with input embeddings and then guided by the powerful spatial semantics of CLIP to achieve spatiotemporal

learning layer by layer. It is worth mentioning that the custom temporal prompts capture long sequence dependencies between frames and the inter-frame communication information between each frame and all other frames to perform temporal modeling.

At the textual level, considering that the semantic supervision information provided by simple original category labels is far from rich and diverse enough, we predefine a Chinese label dictionary and introduce the corresponding Chinese text encoder, and then form joint semantic supervision of Chinese and English together with the CLIP text encoder. The intuition behind it is that Chinese culture is vast and profound, and the profound cultural heritage that Chinese possesses can complement the advantages of other languages. Experiments show that Chinese semantic supervision further improves performance.

We conduct comprehensive experiments on video action recognition datasets. Specifically, under the "few-shot" training setting, we verify the data efficiency and generalization of our method, while under the "closed-set" setting, we compare the accuracy with state-of-the-art techniques. In summary, we make the following contributions:

- We propose a novel method to adapt CLIP to the video domain for efficient action recognition. The method is simple to implement, does not disrupt the original pre-trained parameters, and has very few trainable parameters. Extensive experiments demonstrate the good performance and generalization of our method in various learning settings;
- Taking a visual perspective, we design a temporal prompt reparameterization encoder that aims to enhance the model's temporal modeling capability. The encoder replaces the direct learning style of prompt vectors, allowing the model to learn more generalized temporal representations for specific domains while also being lightweight and efficient;
- At the textual level, we predefine a Chinese label dictionary and introduce the corresponding Chinese text encoder to realize joint semantic constraints of Chinese and English in order to enhance video representations.

2. Related Works

In recent years, significant progress in the field of video action recognition cannot be separated from the development of contrastive visual language pre-training models represented by CLIP and the introduction of parameter-efficient transfer techniques in the video field. Therefore, we will briefly review the relevant work in this field from the following three aspects.

1. Video Action Recognition

The performance of video recognition models depends on their ability to model temporal information. In the early stages, due to the constraints of data and computing power, hand-crafted feature descriptors [4] for spatiotemporal representations have become mainstream; however, they are difficult to design and not easy to generalize. We have witnessed a paradigm shift from CNN-based to Transformer-based methods in deep-learning-dominated methods. Among them, some methods utilize parallel branches [6,27] to jointly model static and dynamic features. 3D convolution has also been widely adopted [28], aiming to directly capture spatiotemporal features. Taking into account the trade-off between efficiency and accuracy, some works [29] decompose convolution in both temporal and spatial dimensions, while other studies [11,30] focus on deploying plug-and-play temporal modules for 2D convolution. Recently, Transformer-based network backbones have been widely used for video recognition [15,31] and are gradually becoming a trend. Considering that most of the above methods are trained in a conventional one-hot supervised manner, recent approaches like ActionCLIP [32] and X-CLIP [33] apply VLMs to action recognition tasks. Given the strong generalization ability of VLMs, these methods perform well in "few-shot" and "zero-shot" learning settings.

2. Vision–Language Pre-trained Models

In recent years, ViT [8] and its various variants [34,35] have shown excellent performance in the image domain, and their pre-trained models exhibit strong generalization capabilities across various downstream tasks. Inspired by this, extensive research works have been conducted on vision–language pre-trained models [10,11], with CLIP [10] being one of the most representative works in this area. Different from conventional training methods, VLMs utilize large-scale image–text pairs crawled from the Internet, with text serving as semantic supervision, and jointly learn fine-grained visual representations through contrastive loss training. Benefiting from the impressive zero-shot transferability demonstrated by VLMs in the image domain, similar ideas are widely applied to various downstream tasks. For example, semantic segmentation tasks [36]; ref. [37] utilizes a pre-trained VLM combined with prompt engineering for object detection; PointCLIP [38] transfers knowledge from CLIP to point cloud understanding tasks. In the video domain, some methods directly replace image–text with video–text, regardless of the cost of pre-training, for tasks like video retrieval [39]. Subsequently, some methods have also carried out work on video recognition using CLIP [17–19,32,33]. One type of work [18,19] still follows the traditional unimodal paradigm, strongly initializing the visual representations learned by CLIP into the backbone network of the video model by discarding the text branch, whereas several works [32,33] follow a full fine-tuning approach to convert the image features in CLIP into video features. By contrast, we directly model temporal cues based on CLIP by fine-tuning only a few additional parameters to effectively adapt the well pre-trained image model to the solution of video tasks, significantly saving training costs due to the simplicity of our proposed approach.

3. Parameter-Efficient Transfer Learning

With the widespread application of large pre-trained language models in various downstream tasks, the idea of efficient tuning [16,25] has been proposed for the first time and has received much attention in the field of NLP, aiming to solve the efficiency and cost problems encountered when fully fine-tuning pre-trained models so as to achieve or exceed the performance of full fine-tuning by updating only a small number of parameters. Some existing methods [40] only train adapters that occupy a small portion of the parameters in the entire model. Some methods exploit sparsity [41] or trainable low-rank decomposition matrices [42] to achieve efficient transfer. Additionally, some methods [25] choose to append some discrete or continuous learnable prompt vectors to the input or intermediate feature sequences of the model, which are used for optimizing specific downstream tasks. Recently, the idea of efficient transfer learning in NLP has been borrowed from the field of computer vision [20–24]. However, the above-mentioned methods mainly focus on fine-tuning models within the same domain; that is, adapting image models to the image domain and video models to the video domain. In this work, we investigate how to adapt image-pre-trained models lacking temporal reasoning ability to video recognition tasks.

This paper is aimed at the problem that most methods in the field of video-action recognition follow the conventional training approach of mapping labels to numbers, lacking the utilization of textual information. We use the VLM learning framework to obtain video representations with rich semantics. As for the problem of fine-tuning between the same domain and fully fine-tuning between cross-domains in previous research on VLMs, we make targeted improvements (see the Section 3 for details).

3. Methods

In this section, we first briefly outline the workflow of VLMs for video action recognition in Section 3.1 and our proposed CLIP-based method framework in Section 3.2, and then elaborate on our two key components, namely video encoder and text encoder, in Sections 3.3 and 3.4, respectively. Finally, the learning objectives of our method are stated in Section 3.5.

3.1. Action Recognition with VLMs

This study is conducted in response to the problem that most studies in the video field predict a set of predefined categories in a one-hot supervised manner within the single-modal framework, making the learned visual representations less universal. In this work, which is inspired by the VLM multi-modal learning framework, we use text labels as supervision signals and jointly train the visual encoder and text encoder to learn how to align the obtained visual representations and corresponding textual representations, thereby extracting rich semantic information.

The workflow of the multi-modal learning framework based on VLMs for video action recognition is shown in Figure 1.

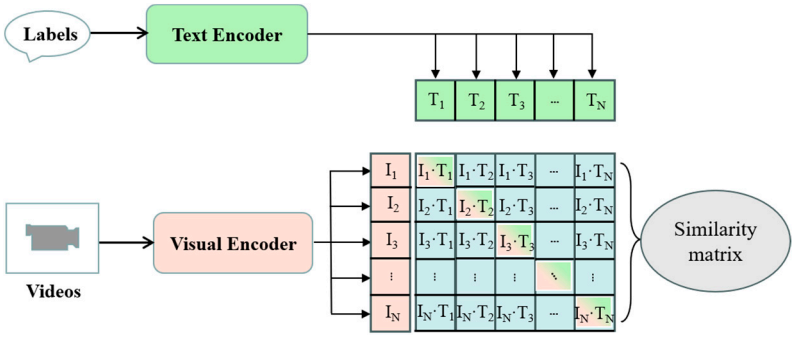


Figure 1. General workflow diagram for VLMs based on contrastive learning.

The general structure of VLMs consists of two separate encoders and a similarity calculation module. Assume that given a set of video samples V composed of multiple static images and a set of category labels, the videos and text labels in the samples are, respectively, encoded by the visual encoder $f(\cdot|\theta_{\mathcal{V}})$ and the text encoder $f(\cdot|\theta_{\mathcal{T}})$ to obtain video embedding representations $\varepsilon\mathcal{V}$ and text label embedding representations $\varepsilon\mathcal{T}$. Using the similarity calculation module, the dense cosine similarity matrix for all video-label pairs is calculated.

3.1.1. Video Representations

Specifically, one video clip $\mathcal{V} \in \mathbb{R}^{T\times H\times W\times 3}$ is selected from the set of samples, which consists of T frames with a spatial resolution of $H \times W$. Following the ViT architecture, each image frame is segmented into N non-overlapping square patches of spatial size $P \times P$, where the total number of patches is $N = H/P \times W/P$. Then, all square patches$\in \mathbb{R}^{P\times P\times 3}$ of each frame are flattened into a set of vectors that are then projected through a linear projection layer to generate patch embeddings represented as $[x_{t,1}^{(0)}, x_{t,2}^{(0)}, \ldots, x_{t,N}^{(0)}]$, where $t = \{1, 2, \ldots, T\}$ represents the frame number and the superscript represents the number of the encoder layer where the feature is located. Subsequently, an additional learnable classification token x_{cls} is prepended to the patch embedding sequences for each frame. Finally, the frame sequence input of each ViT block of the visual encoder is given by Equation (1):

$$\begin{aligned} Z^{(l)} &= [z_1^{(l)}, \ldots, z_T^{(l)}], \\ z_t^{(l)} &= [x_{cls}^{(l)}, x_{t,1}^{(l)}, x_{t,2}^{(l)}, \ldots, x_{t,N}^{(l)}] + e^{\text{spatial}} + e^{\text{temporal}} \end{aligned} \tag{1}$$

where $l \in \{1, \ldots, L\}$, e^{temporal}, and e^{spatial} denote temporal and spatial position encoding, respectively, and $(N + 1)$ embedded sequences are added element-by-element with them to obtain spatiotemporal enhancement. In the L-layer ViT block of the visual encoder, the frame-level embedding of the *t-th* frame feature of the *i-th* layer is described using Equation (2):

$$z_t^{(i)=} f\left(z_t^{(i-1)} \middle| \theta\mathcal{V}\right)^{(\mathrm{i})} \in \mathbb{R}^{(N+1)\times D} \tag{2}$$

where $i \in \{1, \ldots, L\}$ refers to the Transformer block layer index and D is the channel dimension.

Then, the classification token x_{cls} is extracted from the output embeddings of the last layer $z_t^{(L)}$ and mapped to the $\widetilde{D}$ dimension through a linear projection layer $Pproj \in \mathbb{R}^{D \times \tilde{D}}$ to obtain the final frame-level representation sequences, as in Equation (3):

$$\begin{aligned} F &= [F_1, \ldots, F_t, \ldots, F_T], \\ F_t &= \mathrm{P}_{proj}^{\mathrm{T}} z_{t,0}{}^{(L)} \in \mathbb{R}^{\tilde{D}} \end{aligned} \tag{3}$$

where F_t is the frame-level representation of frame t and $z_{t,0}{}^{(L)}$ represents the classification token x_{cls} in the output embeddings of the *t-th* frame in the last layer of the visual encoder. Finally, in order to obtain the video-level representations, the extracted frame-level representation sequences F is generally averaged by pooling, which is expressed as Equation (4).

$$\varepsilon\mathcal{V} = \mathrm{MeanPool}(F) \tag{4}$$

3.1.2. Text Representations

The text representations $\varepsilon\mathcal{T} = f(\mathcal{T}|\theta_{\mathcal{T}})$ is generated by the text encoder for the input text label $\mathcal{T}$. Considering that the text description information during VLM pre-training is basically short sentences, and that most of the labels in existing datasets exist in the form of words, in order to compensate for this difference in data distribution and solve the problem of polysemy, it is common to convert a word into a form like "A video of the action of {label}" via manual prompt templates [10]. However, manual prompt templates are not static, and appropriate prompt templates should be selected for specific datasets.

3.1.3. Similarity Calculation

Finally, the visual encoder and the text encoder based on VLMs obtain the video representations $\varepsilon\mathcal{V}$ and the text representations $\varepsilon\mathcal{T}$, and the similarity score $S_{\mathcal{V}\leftrightarrow\mathcal{T}}$ between the two is calculated using the cosine similarity function, represented as Equation (5):

$$S_{\mathcal{V}\leftrightarrow\mathcal{T}} = s(\varepsilon\mathcal{T}, \varepsilon\mathcal{V}) \tag{5}$$

where $s(\cdot, \cdot)$ denotes the cosine similarity function. During training, the objective is to maximize the scores of matched video-label pairs while minimizing the scores of all other mismatched pairs. During inference, the scores between the input video and each category label are calculated and sorted as the prediction result.

3.2. Proposed CLIP-Based Framework

We present our framework in Figure 2. Unlike the CLIP model, which requires two parallel encoders to generate visual and textual representations separately, our proposed model consists of three encoders. In this paper, we extend the CLIP visual encoder in a lightweight manner and introduce an additional text encoder, aiming to transfer VLMs to action recognition tasks, as described in Sections 3.3 and 3.4.

Different from the conventional method of obtaining video representations $\varepsilon\mathcal{V}$ by mean pooling the frame-level representations F, we use word embedding as a query to calculate the similarity between each word and T frames, further perform softmax to obtain the similarity score between 0 and 1, and finally aggregate the similarity scores between specific frames and different words to obtain the weight coefficients for each frame. The specific expression is described by Equation (6):

$$W_t = \frac{1}{N} \sum_{n=1}^{N} \frac{\exp(F_t^{\mathrm{T}} w_n / \tau)}{\sum_{t=1}^{T} \exp(F_t^{\mathrm{T}} w_n / \tau)}, t \in [1, T], n \in [1, N] \tag{6}$$

where w_n represents the word embeddings obtained by the text encoder, N represents the number of words in the category label, and τ is the temperature hyperparameter. Next, we aggregate these frame-level representations using the weight coefficients of each frame to obtain the final enhanced video representations $\varepsilon\mathcal{V}$, as in Equation (7):

$$\varepsilon\mathcal{V} = \sum_{t=1}^{T} F_t \cdot W_t \tag{7}$$

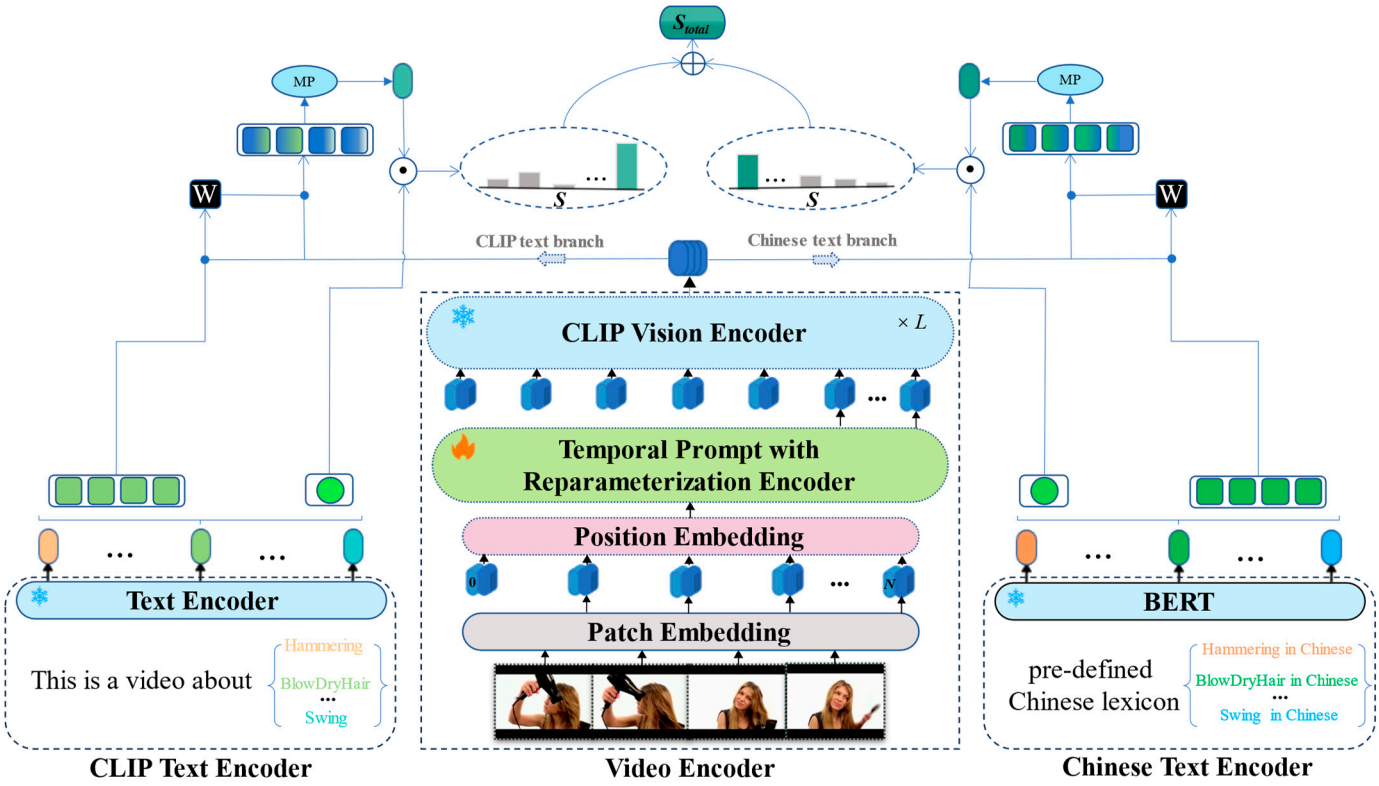


Figure 2. Overview of the overall framework. Our proposed method contains three branches: video encoder, Chinese text encoder, and CLIP text encoder.

3.3. Video Encoder

At the visual level, our proposed video encoder is roughly divided into two parts: (1) a simple yet effective temporal prompt reparameterization module, which re-encodes the received temporal prompt vectors into more discriminative and adaptive temporal representations; (2) the frozen original image encoder from CLIP, which provides semantic guidance for spatiotemporal modeling with its powerful spatial semantics.

3.3.1. Temporal Prompts

Our custom prompts are shown in Figure 3. Formally, we first extract the classification tokens of all frames in the previous layer, perform linear projection P*cls* on them, and then apply multi-head self-attention to obtain the dependencies of the current frame with other frames, represented by Equation (8):

$$\begin{aligned}
&\mathrm{X}_{cls}^{(l-1)} = [\mathrm{P}_{cls}^{\mathrm{T}}\mathrm{z}_{1,0}^{(l-1)}, \ldots, \mathrm{P}_{cls}^{\mathrm{T}}\mathrm{z}_{t,0}^{(l-1)}, \ldots, \mathrm{P}_{cls}^{\mathrm{T}}\mathrm{z}_{T,0}^{(l-1)}] \in \mathbb{R}^{T\times 1\times D},\\
&\mathrm{C}^{(1)} = \mathrm{MHSA}(\mathrm{LN}(\mathrm{X}_{cls}^{(l-1)})) + \mathrm{X}_{cls}^{(l-1)},\\
&\mathrm{X}_{cls}^{(l-1)} = [\hat{\mathrm{x}}_{1,cls}^{(l-1)}, \cdots, \hat{\mathrm{x}}_{t,cls}^{(l-1)}, \cdots, \hat{\mathrm{x}}_{T,cls}^{(l-1)}],\\
&\mathrm{C}^{(l)} = \left[\ c_1^{(l)}, \ldots, c_T^{(l)}\ \right]
\end{aligned} \tag{8}$$

where $C^{(l)}$ represents the classification token sequences calculated by attention in the *l-th* layer, MHSA represents the multi-head self-attention operation, and LN represents layer normalization.

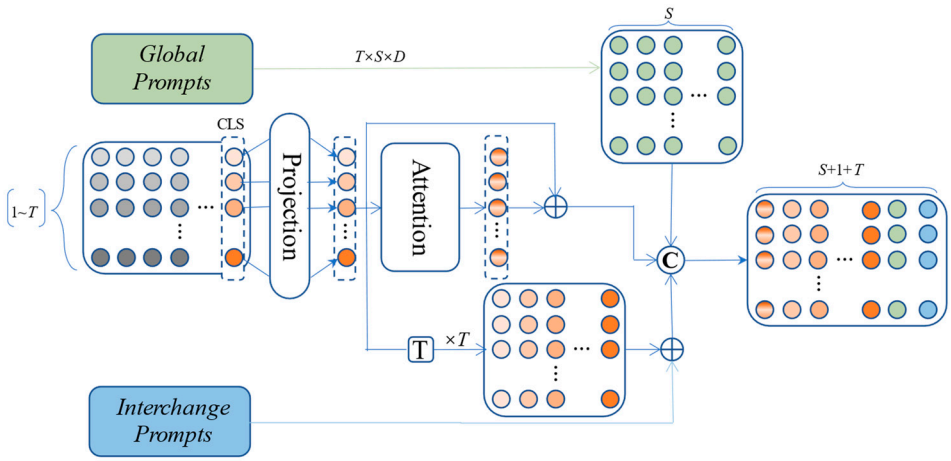


Figure 3. An illustration of temporal prompts. The final temporal prompts consist of global prompts, interchange prompts, and CLS tokens mapped with the attention module.

Next, we introduce the randomly initialized learnable global prompts $G^{(l)} = [{g_1}^{(l)}, \ldots, {g_s}^{(l)}] \in \mathbb{R}^{T\times S\times D}$, which are used to provide the model with additional long-sequence dependencies of the video. Here, S represents the number of global prompts, which can be freely adjusted.

Subsequently, we combine randomly initialized interchange prompts $I^{(l)} = [{i_1}^{(l)}, \ldots, {i_T}^{(l)}]$ with the previously extracted classification token $X_{cls}^{(l-1)}$ representing the current frame information, thereby obtaining discriminative information between each frame and all other frames, represented by Equation (9):

$$\begin{aligned} \hat{X}_{cls}^{(l-1)} &= (X_{cls}^{(l-1)})^{\mathrm{T}} T \in \mathbb{R}^{T\times T\times D}, \\ \hat{I}^{(l)} &= I^{(l)} + \hat{X}_{cls}^{(l-1)} \end{aligned} \tag{9}$$

Finally, the above prompts are concatenated to obtain the final temporal prompts, as in Equation (10):

$$\mathcal{P} = \left\{ C^{(l)}, G^{(l)}, \hat{I}^{(l)} \right\} \in \mathbb{R}^{T\times P\times D} \tag{10}$$

For convenience, we simplify it to the format represented by Equation (11):

$$\mathcal{P} = [\mathcal{P}_1][\mathcal{P}_2]\ldots[\mathcal{P}_H], P = T + 1 + S \tag{11}$$

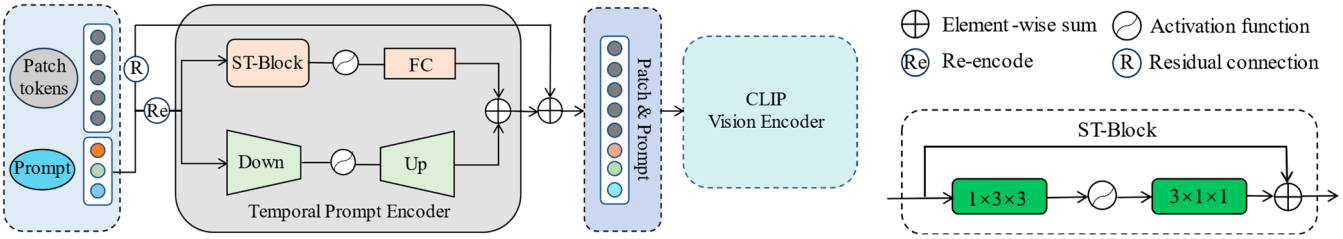
3.3.2. Reparameterization Encoder

Taking inspiration from the effectiveness of language model reparameterization [25,26], we propose a lightweight reparameterization encoder. This encoder is fine-tuned on the downstream task such that prompt tokens undergo domain-specific adjustments before being forwarded to the fixed visual encoder to better provide adaptive temporal modeling for video tasks. Specifically, we project the temporal prompts $\mathcal{P}$ as reparameterized embedding sequences $\mathcal{P}_R$ via the encoder, as in Equation (12):

$$\mathcal{P}_R = R(\mathcal{P}) = \beta(\mathcal{P}) + \mathcal{P} \tag{12}$$

where $R(\cdot)$ represents the reparameterized function of the encoder, which consists of a network $\beta(\cdot)$ with residual connection, as shown in Figure 4a. The ST-Block in the network $\beta(\cdot)$ draws on the design idea in the R(2+1)D model [29] by default to further enhance spatiotemporal modeling; that is, decomposes the 3D space–time convolution into 2D

spatial convolution and 1D temporal convolution, as shown in Figure 4b. Meanwhile, there are other designs available for ST-Block, which are explored in the Ablation Studies Section. The network $\beta(\cdot)$ implements task-related reparameterization in prompt embeddings, aiming to enhance domain-specific temporal information in prompts and improve the model's adaptability. Different encoder network architectures have varying impacts on the model's adaptability and performance, which are explored in the Ablation Studies Section. Additionally, the residual connection in the encoder enables the model to more flexibly combine the original embeddings of prompts with the embeddings projected from the network, thus integrating the raw knowledge encoded in CLIP parameters with new learned knowledge obtained from training samples through the network $\beta(\cdot)$, as shown in Equation (12). The Ablation Studies Section provides the effect of residual connection in the network $\beta(\cdot)$ on the model.



(**a**) Reparameterization Encoder (**b**) ST-Block in $\beta(\cdot)$.

Figure 4. (**a**) Shows the structural details of the reparameterization encoder $\beta(\cdot)$; (**b**) illustrates the detailed information on ST-Block in $\beta(\cdot)$.

Overall, the core idea of our proposed reparameterization encoder is to reparameterize the prompts embeddings before forwarding them to the CLIP visual encoder with fixed parameters, rather than letting the model directly optimize the prompts during back-propagation optimization. Through the learnable reparameterization encoder, our model can flexibly capture task-related dependencies within the prompts embeddings, freeing itself from the limitation of being encoded only by the frozen CLIP vision encoder. In addition, combining these associate-learned prompt tokens that have undergone reparameterization with the input tokens through concatenation operation and feeding them into the CLIP visual encoder will be more conducive to the model learning more contextually generalizable and field-specific knowledge during the optimization process.

3.3.3. Spatial Semantic Guidance of CLIP

Finally, the projected context sequences $\mathcal{P}_R$ obtained from the encoder is attached to the $Z^{(l-1)}$ embedding sequences and applied to the multi-head self-attention in the frozen CLIP visual encoder of this layer, as in Equation (13):

$$[\hat{Z}^{(l)}, \mathcal{P}_R] = \text{MHSA}(\text{LN}([Z^{(l-1)}, \mathcal{P}_R])) + [Z^{(l-1)}, \mathcal{P}_R] \tag{13}$$

Before feeding $\hat{Z}^{(l)}$ into the multi-layer perceptron (MLP) in the CLIP visual encoder of this layer, we remove the additional prompt tokens for subsequent processing, as in Equation (14):

$$Z^{(l)} = \text{MLP}(\text{LN}(\hat{Z}^{(l)})) + \hat{Z}^{(l)} \tag{14}$$

Then, the subsequent $Z^{(l)}$ obtained will repeat the above operation in the next layer until $Z^{(L)}$ is obtained.

3.4. Text Encoder

At the text level, we introduce an additional text encoder pre-trained with Chinese text pairs, aiming to combine with the CLIP text encoder to form joint supervision of Chinese and English semantics to strengthen text constraints so as to obtain stronger video

representations. Specifically, to bridge the text format gap between CLIP pre-training data and downstream datasets, we append the original labels with the prefix prompt "This is a video about {label}" before feeding them into the CLIP text encoder. For the Chinese text BERT encoder $f(\cdot|\theta_{\mathcal{C}})$ introduced from the Chinese-CLIP model [43], we adopt a predefined Chinese label dictionary as its input. We believe that the rich semantics contained in Chinese play a key role in obtaining more robust video representations.

3.5. Learning Objectives

To sum up, our architecture extracts embedded representations of videos, predefined Chinese labels, and category labels $\varepsilon\mathcal{V}$, $\varepsilon\mathcal{C}$, $\varepsilon\mathcal{T}$ through corresponding encoders, all of which are initialized by the weights of the pre-trained model. In this paper, we freeze the original encoder and only fine-tune the additional module to apply the image representations generated by the VLM to the video task.

During training, our objective is to maximize the similarity between $\varepsilon\mathcal{V}$ and $\varepsilon\mathcal{T}$ when they are correlated and minimize the similarity when they are uncorrelated. $\varepsilon\mathcal{V}$ and $\varepsilon\mathcal{C}$ are the same. Formally, assume a batch of B quadruples $\{\varepsilon\mathcal{V}i, \varepsilon\mathcal{T}i, \varepsilon\mathcal{C}i \equiv C[y_i], y_i\}_{i=1}^{B}$, where C is a set of K categories, with subscripts $y_i \in [0, K-1]$ representing the index of labels in the dataset, and $\varepsilon\mathcal{V}i, \varepsilon\mathcal{T}i, \varepsilon\mathcal{C}i$ denoting the video embedding, category label embedding, and Chinese label embedding of the *i-th* video, respectively. We consider the learning objectives as per the conventional practice [19,32], where the loss function uses symmetric cross-entropy to maximize the similarity between $\varepsilon\mathcal{V}$ and $\varepsilon\mathcal{T}$ when matched and minimize the similarity between other irrelevant pairs, formulated as in Equation (15):

$$\begin{aligned}
\mathcal{L}_{\mathcal{V}\to\mathcal{T}} &= -\frac{1}{B}\sum_{i}^{B}\frac{1}{|\mathcal{K}(i)|}\sum_{k\in\mathcal{K}(i)}\log\frac{\exp(s(\varepsilon\mathcal{T}i, \varepsilon\mathcal{V}k)/\tau)}{\sum_{j}^{B}\exp(s(\varepsilon\mathcal{T}i, \varepsilon\mathcal{V}j)/\tau)}, \\
\mathcal{L}_{\mathcal{T}\to\mathcal{V}} &= -\frac{1}{B}\sum_{i}^{B}\frac{1}{|\mathcal{K}(i)|}\sum_{k\in\mathcal{K}(i)}\log\frac{\exp(s(\varepsilon\mathcal{T}k, \varepsilon\mathcal{V}i)/\tau)}{\sum_{j}^{B}\exp(s(\varepsilon\mathcal{T}j, \varepsilon\mathcal{V}i)/\tau)}, \\
\mathcal{L}_{\mathcal{V}\rightleftarrows\mathcal{T}} &= \frac{1}{2}(\mathcal{L}_{\mathcal{V}\to\mathcal{T}} + \mathcal{L}_{\mathcal{T}\to\mathcal{V}})
\end{aligned} \tag{15}$$

where $k \in \mathcal{K}(i) = \{k|k \in [1, B], y_k = y_i\}$ and τ generally refers to the temperature hyperparameter used for scaling. Similarly, the loss between $\varepsilon\mathcal{V}$ and $\varepsilon\mathcal{C}$ is calculated as in Equation (16):

$$\begin{aligned}
\mathcal{L}_{\mathcal{V}\to\mathcal{C}} &= -\frac{1}{B}\sum_{i}^{B}\frac{1}{|\mathcal{K}(i)|}\sum_{k\in\mathcal{K}(i)}\log\frac{\exp(s(\varepsilon\mathcal{C}i, \varepsilon\mathcal{V}k)/\tau)}{\sum_{j}^{B}\exp(s(\varepsilon\mathcal{C}i, \varepsilon\mathcal{V}j)/\tau)}, \\
\mathcal{L}_{\mathcal{C}\to\mathcal{V}} &= -\frac{1}{B}\sum_{i}^{B}\frac{1}{|\mathcal{K}(i)|}\sum_{k\in\mathcal{K}(i)}\log\frac{\exp(s(\varepsilon\mathcal{C}k, \varepsilon\mathcal{V}i)/\tau)}{\sum_{j}^{B}\exp(s(\varepsilon\mathcal{C}j, \varepsilon\mathcal{V}i)/\tau)}, \\
\mathcal{L}_{\mathcal{V}\rightleftarrows\mathcal{C}} &= \frac{1}{2}(\mathcal{L}_{\mathcal{V}\to\mathcal{C}} + \mathcal{L}_{\mathcal{C}\to\mathcal{V}})
\end{aligned} \tag{16}$$

In summary, the total loss $\mathcal{L}$ expression is described by Equation (17):

$$\mathcal{L} = \mathcal{L}_{\mathcal{V}\rightleftarrows\mathcal{T}} + \mathcal{L}_{\mathcal{V}\rightleftarrows\mathcal{C}} \tag{17}$$

In the inference stage, we use Equation (18) to combine the similarity scores between the video and two types of labels to obtain the final inference result.

$$\mathcal{S} = \lambda\mathcal{S}_{\mathcal{V}\leftrightarrow\mathcal{T}} + (1-\lambda)\mathcal{S}_{\mathcal{V}\leftrightarrow\mathcal{C}} \tag{18}$$

where λ is the fusion weight.

4. Experiments

In this section, we conduct experiments on three widely used video datasets, HMDB-51, UCF-101, and Something-Something V1, under different settings, namely fully supervised and few-shot. The three video datasets cover a broad range of activities, and such diverse datasets allow us to comprehensively evaluate the model across different scales and domains. We first introduce a series of environmental configurations and implementation details used in the experiments, then conduct ablation research on the key components of our proposed method. Extensive experiments demonstrate the efficiency and generalization of our method on "few-shot" training, as well as the Top-1 accuracy of the best model on the three datasets, which is competitive with most existing state-of-the-art methods.

4.1. Experimental Configurations and Details

In this section, we briefly describe the datasets required for the experiment, a series of environment configurations and implementation details for the model, and the experimental baseline.

4.1.1. Datasets and Evaluation

We evaluate the proposed method on three different benchmarks. Specifically, HMDB51 is a small dataset that provides about 7 K videos of 51 action categories. We use all three splits; each split consists of 3570 and 1530 videos for training and evaluation, respectively. The UCF101 dataset contains a total of 13,320 video samples, divided into 101 different action categories. The videos cover various activities and actions in real life, and each video has a duration ranging from 7 s to 20 s. Something-Something V1 (SSv1) is a more challenging dataset since it requires more temporal modeling. It contains 174 fine-grained human activities, with a total of approximately 86 K/12 K training and validation videos. Details about the datasets are given in Table 1.

Table 1. Detailed information on HMDB-51 and UCF-101 datasets.

Datasets	Categories	Training Set	Test Set	Total	Split	Sources
HMDB-51	51	3570	1530	7000	3	Movies and web videos
UCF-101	101	9537	3783	13,320	3	YouTube
Something-Something V1	174	∼86 K	∼12 K	∼100 K	1	Crowdsourcing collection

Unless otherwise stated, we spatially scale the short side of each input frame in the datasets to 256, and the input frame resolution is 224 × 224. To balance inference speed and accuracy, we adopt two evaluation protocols: (1) Single view: we efficiently evaluate each video using 1 central crop and clip; (2) Multi-view: to improve model accuracy, it is routine to randomly sample each video to obtain multiple clips with multiple spatial crops [6]. To achieve optimal performance, we use 4 temporal clips × 3 spatial crops, and the final Top-1 and Top-5 accuracies are derived from the average score of all views.

4.1.2. Training Details

In our experiment, we used CLIP [10] pre-trained image and text encoders for visual and original text label level processing, while for the encoding predefined Chinese labels, we chose the RoBERTa-wwm-Base architecture in the Chinese-CLIP [43] as the Chinese text encoder, and kept all encoders' pre-trained parameters constant during training. In the training phase, we set all temperature hyperparameters τ to 0.01 and input frames T (e.g., 8, 16, 32) based on the sparse sampling strategy. Details of the remaining hyperparameters are shown in Table 2.

Table 2. Detailed settings of model training hyperparameters.

Setting	Value
Training Hyperparameter	
Batch size	256 (Fully), 64 (Few-shot)
Training epochs	30 (ViT-B), 20 (ViT-L)
Optimizer	AdamW, betas = [0.9,0.999]
learning rate	5×10^{-6} (Fully), 4×10^{-6} (Few-shot)
Learning rate schedule	cosine
Linear warm-up epochs	5
Weight decay	1×10^{-2}
Data Augmentation	
Training resize	RandomSizedCrop
Training crop size	224
Random Flip	0.5
Gray Scale	0.2

4.1.3. Baseline

To analyze the effectiveness of each component in our method, we design a "baseline" model for ablation research. This model is based on CLIP and only replaces mean pooling with aggregating frame-level representations using the weight coefficients of each frame to obtain the final video-level representations, as shown in Equation (7). Compared with our method, there are two differences: (1) at the visual level, the input embeddings before the CLIP image encoder do not contain the temporal prompts obtained by the reparameterization encoder; (2) at the textual level, the predefined Chinese label text encoder branch is not applied to the "baseline".

4.2. Ablation Studies

In this section, we provide detailed ablation studies to illustrate the effectiveness of our proposed key design. We train all models on 2 NVIDIA GeForce RTX 4090 GPUs, unless otherwise stated; the models in this section use ViT-B/16 as the backbone, and all experiments are trained on the HMDB-51 and UCF-101 training sets and tested on the validation sets using a single view. The results are shown in Figure 5.

4.2.1. Effect of Key Components

In Figure 5a, we obtain video representations through mean pooling over 8 frames to evaluate the performance of the original CLIP model on the video datasets. Compared to the "baseline", our model significantly improves the Top-1 accuracy by equipping only temporal prompts, increasing by +5.5% and +8.2% on the HMDB-51 and UCF-101 datasets, respectively. After further reparameterizing the temporal prompts through the $\beta(\cdot)$ network to obtain domain-specific temporal information, model performance on the two datasets is further improved by +3.5% and +2.7%, respectively. It is evident that temporal modeling plays a crucial role in bridging the significant gap between image and video domains. Moreover, the additional Chinese label text encoder branch also shows positive effects, with the Top-1 accuracy improved by +2.9% and +3.1% on the HMDB-51 and UCF-101 datasets, respectively, indicating the effectiveness of the rich semantics contained in Chinese for obtaining more powerful video representations.

4.2.2. Number of Sampled Frames

We explored the impact of the number of sampled video frames T on the model in Figure 5b. We compared multiple results for $T \in \{4, 8, 16, 32\}$, and observed that the performance gain on HMDB-51 gradually levels off and that on UCF-101 declines as the number of T increases. Although the model with T equal to 32 performs the best on HMDB, we believe that the corresponding increase in computational cost can be completely negligible given the slight performance gain. To balance performance and efficiency, we set T to 16 in the subsequent experiments and we speculated that the reason why the

model cannot obtain proportional performance gains from more frames is mainly due to the existence of redundant frames that contain a lot of background noise unrelated to the task, which cannot be eliminated by additional parameters. As a result, the frozen CLIP image encoder introduces internal variance between frames, thus not achieving the expected results.

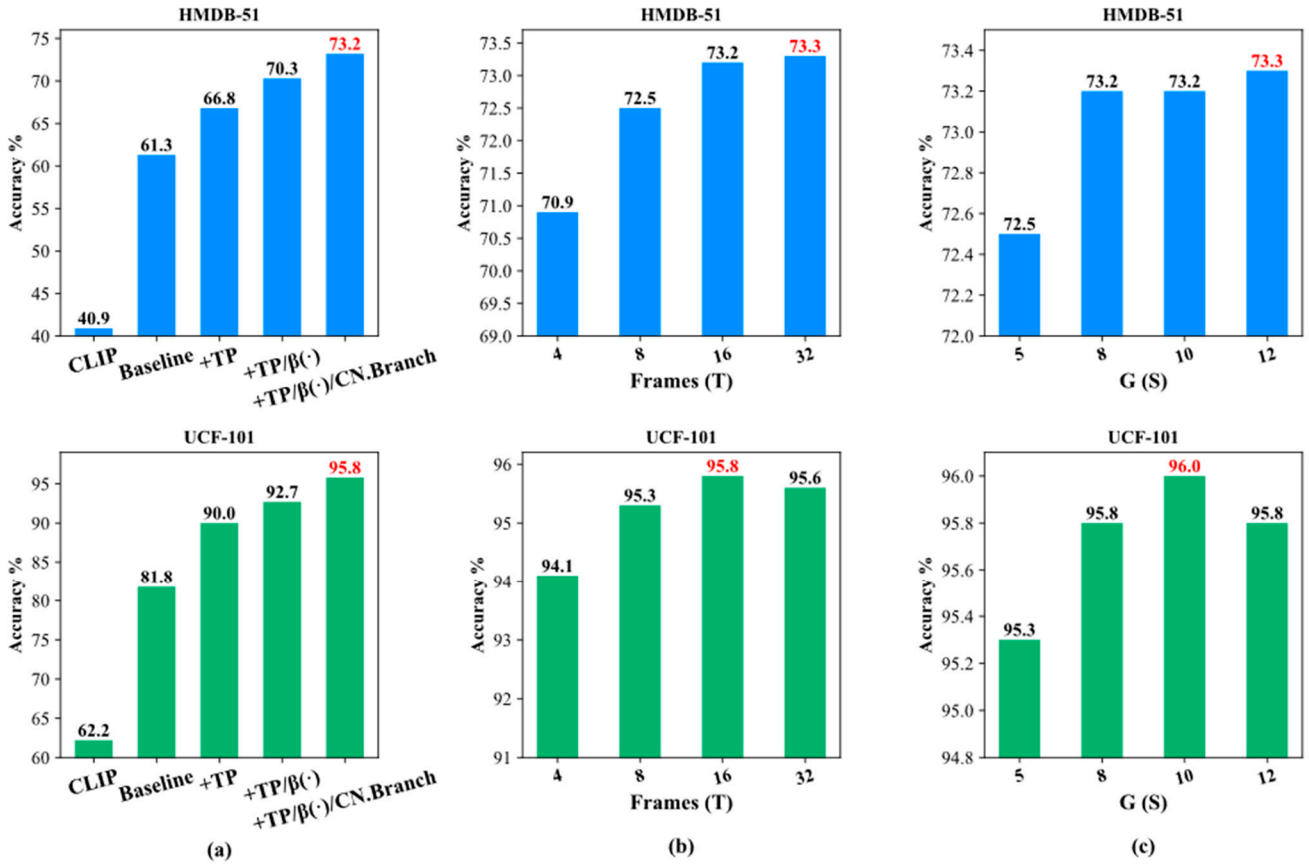


Figure 5. Ablation research using ViT-B/16 as the backbone on the HMDB-51 and UCF-101 datasets. The experiments in this chapter all use single-view testing. (**a**) The effectiveness of our proposed key components is proved step by step; (**b**) illustrates the impact of the number of sampling frames on the model; (**c**) shows the performance of learnable global prompts with different lengths. Note: TP is the abbreviation for Temporal Prompts; CN.Branch is the abbreviation for Chinese Label Text Encoder Branch.

4.2.3. Length of Learnable Global Prompts

As shown in Figure 5c, we evaluated the impact of global prompts on model performance when $S \in \{5, 8, 10, 12\}$ under the same settings. The results show that as S increases, the performance gain of the model is gradually improved, except when S is 12. When S increases from 5 to 12, the Top-1 accuracy of our model increases by 0.8% and 0.5% on HMDB-51 and UCF-101, respectively. We speculate that longer global prompts provide the model with a clearer and more concise overall overview of the sampled frames. When S equals 8, the accuracy of the model almost tends to saturation. Therefore, we set S to 8 by default in the subsequent experiments.

4.2.4. The Influence of Different $\beta(\cdot)$ Networks

To further investigate the impact of different reparameterization network architectures on model performance, we implemented a series of structural variants for the $\beta(\cdot)$ network, including (i) Bottleneck MLP: composed of a down-projection layer, activation function, and an up-projection layer; (ii) Shallow Transformer network, consisting of two Transformer encoders; and (iii) Long Short-Term Memory (LSTM) [44] network. The ST-Block in the $\beta(\cdot)$ network we adopt by default is designed to provide reparameterization of spatiotemporal information and offers several designs to choose from, such as convolution-based TAdaConv [45], C3D [46], and R(2+1)D [29]. The Top-1 and Top-5 accuracies of the aforementioned networks on HMDB-51 and UCF-101 are shown in Table 3. The results show that our proposed $\beta(\cdot)$ network is more conducive to reparameterizing the prompt vectors to adapt VLMs to video tasks, with the ST-Block based on R(2+1)D achieving the best performance. In contrast, the performance of the Bottleneck MLP network is slightly inferior to those of other networks, ultimately due to the fact that MLP is a feed-forward network that lacks capturing valuable contextual information in prompt vectors.

Table 3. The impact of different structural variants of $\beta(\cdot)$ network on model performance. "-" indicates that this module is not available for $\beta(\cdot)$.

$\beta(\cdot)$	TAda	C3D	R(2+1)D	HMDB-51		UCF-101	
				Top-1	Top-5	Top-1	Top-5
MLP		-		70.1	91.3	92.6	96.2
LSTM				71.0	91.8	93.9	97.0
Transformer Encoders				71.8	92.5	94.4	97.5
Ours	✓			72.3	92.9	94.8	98.2
		✓		72.8	93.5	94.6	98.2
			✓	73.2	93.6	95.8	99.0

4.2.5. The Role of Residual Connection in $\beta(\cdot)$

We conducted ablation studies on different ST-Block designs in the $\beta(\cdot)$ network to evaluate the impact of residual connection in the $\beta(\cdot)$ network on performance. Specifically, we compared the performance of structural variants when prompt vectors pass through the $\beta(\cdot)$ network without additional residual connection. As shown in Table 4, removing the residual connection from the $\beta(\cdot)$ network results in varying degrees of decrease in Top-1 and Top-5 accuracies of all structural variants on HMDB-51 and UCF-101, indicating significant vulnerabilities in the network when handling input prompt embeddings. We speculate that the removal of residual connection diminishes the model's ability to capture complementary knowledge, thereby weakening the model's performance.

Table 4. Ablation study of different ST-Block designs in the $\beta(\cdot)$ network with and without residual connection.

$\beta(\cdot)$		Residual Connection	HMDB-51		UCF-101	
			Top-1	Top-5	Top-1	Top-5
Ours	TAda	×	70.5	92.0	95.1	97.8
		✓	72.3	92.9	94.8	98.2
	C3D	×	71.2	92.5	92.9	97.4
		✓	72.8	93.5	94.6	98.2
	R(2+1)D	×	70.0	90.9	93.0	96.0
		✓	73.2	93.6	95.8	99.0

4.2.6. Hand-Crafted [CLS] Prefix Prompt

We explored the initialization effect of attaching different prefix prompts to text labels. We designed three manually crafted prompts with lengths of 3, 4, and 5. As shown in Table 5, the performance of the model was hardly affected by manual prompts for different initializations, proving that the manual prefix we adopted is relatively stable for video tasks.

Table 5. The initialization effects of different prefixes for [CLS].

Manual Prefix Initialization for [CLS]	HMDB-51		UCF-101	
	Top-1	Top-5	Top-1	Top-5
a video about [CLS].	73.1	93.7	95.8	98.8
This is a video about [CLS].	73.2	93.6	95.8	99.0
the video is about [CLS].	72.9	93.6	95.6	98.9

4.2.7. Trainable Parameters and Time Efficiency

Table 6 shows the comparison between our method and the CLIP-based methods under the same hardware and backbone network. When the input frames are 8 or 16, our method outperforms other methods, except for BIKE; however, compared to BIKE, our method is slightly better when T = 8 on HMDB. We also report the comparison of trainable parameter occupancy with Vita, BIKE, and XCLIP in Figure 6a, where our trainable parameters account for only 50% and 40% of those of BIKE and XCLIP, respectively (that is, 54.2 M vs. 106.8 M/131.5 M). In addition, in Figure 6b, we show that our proposed method still performs well when reducing training cycles, even at lower training costs.

Table 6. Comparison of trainable parameters and training time and memory between our method and the CLIP-based method on the same hardware and backbone network. Note: The units for tunable parameters and memory are both MB.

Methods	Backbone	*T*	Tunable Parameters	Epoch	Batch Size	Training GPU Minutes (HMDB)	Memory (HMDB)	Top-1 (%)	
								HMDB	UCF
Vita	ViT-B/16	8	38.88	30	96	45	17,721	67.25	91.54
		16			48		18,182	63.79	90.38
BIKE		8	106.8 (+100%)		32	24	6445	72.22	96.15
		16				45	10,603	73.31	96.63
XCLP		8	131.5 (+147%)		16	32	17,674	70.22	94.20
		16			8	54		70.75	94.10
Ours		8	54.2		32	22	6601	72.50	95.33
		16				41	10,759	73.21	95.81

4.2.8. Visualization

The visualization effect of the attention map in our method is shown in Figure 7. We compared our method with the baseline that does not include our proposed temporal prompts and reparameterized encoder. It can be observed that our proposed approach is more conducive for the model to focus on the dynamic regions and key parts used for the final recognition task in the video.

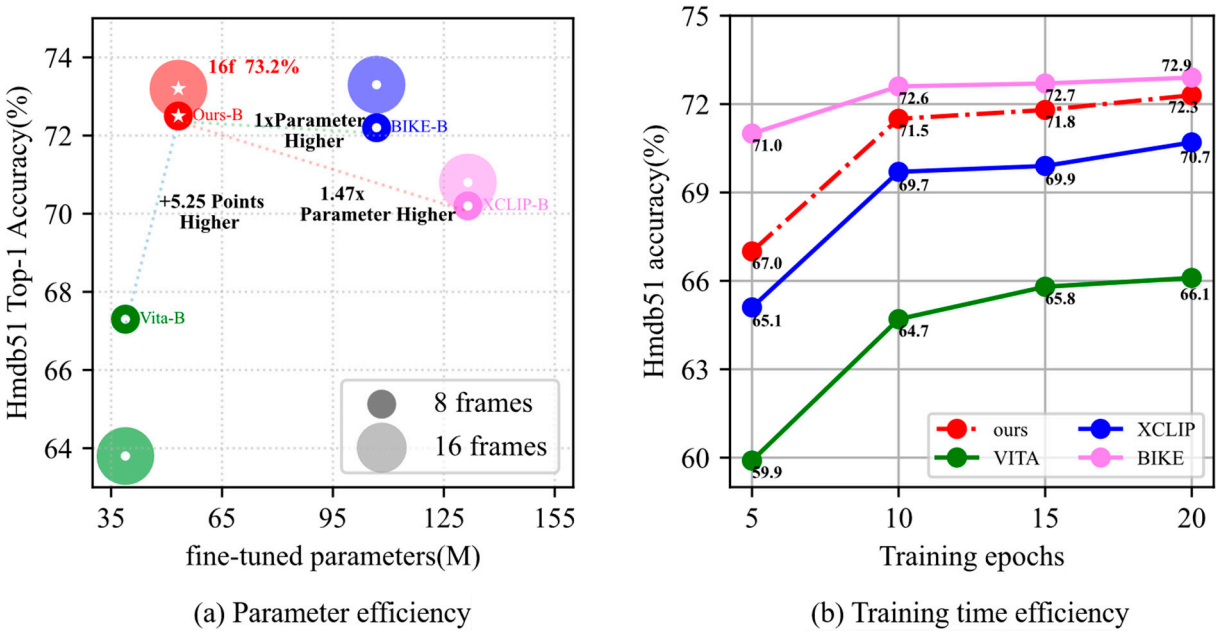


Figure 6. Ablation study of training efficiency. (**a**) Parameter efficiency. Comparison of the number of trainable parameters between our method and Vita, which also freezes the visual backbone, as well as BIKE and XCLIP, which fine-tune the visual backbone; (**b**) training time efficiency. Our model still performs well with fewer training epochs. Note: the asterisk in Figure (**a**) only represents our method and has no special meaning, while the circle represents other methods.

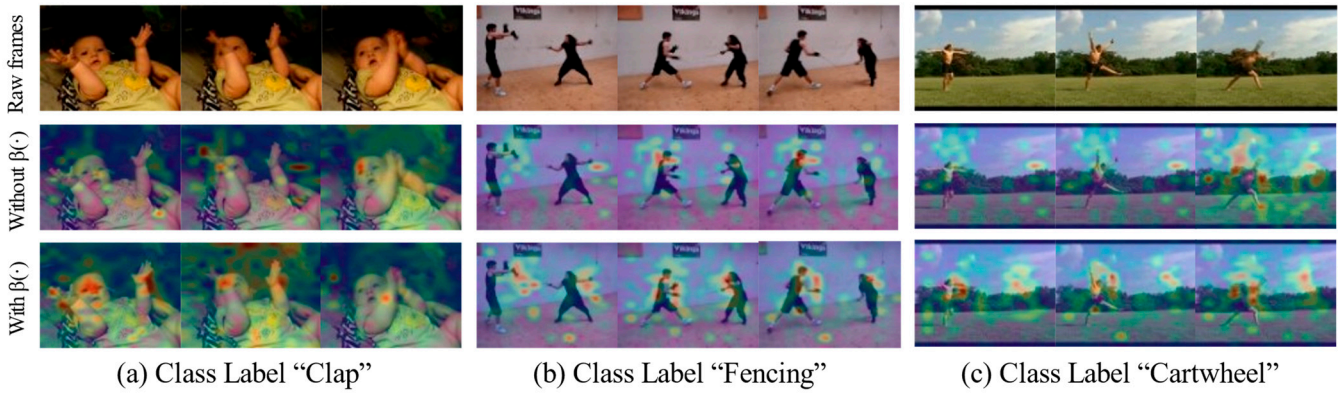


Figure 7. We illustrate some behavioral actions such as "Clap", "Cartwheel", and "Fencing" on the original video frames and the attention map without and with our proposed method. It can be observed that our method focuses on distinguishable motion information and some key regions. Note: red indicates the areas that the model focuses on, while green mainly represents the background or some less important areas.

4.3. Few-Shot Video Recognition

In this section, we conduct few-shot experiments on HMDB-51 and UCF-101 datasets to demonstrate the few-shot recognition capability of our method, which achieves video recognition using only a small number of training samples. For a fair comparison, we follow the standard K-shot setting as in X-CLIP [33]; that is, randomly select K samples from each category to construct a training dataset. Here, we set $K \in \{2, 4, 8, 16\}$ and evaluate the "few-shot" model using single-view on a standard test set. The final Top-1 accuracy is the average of multiple inference results. We train our model using CLIP pre-trained ViT-B/16 without further pre-training on Kinetics-400, with a batch size of 64 and an initial learning rate of 4×10^{-6}. We not only compare the "few-shot" capability of our model

with state-of-the-art and representative methods but also explore the performance of key components we proposed under the "few-shot" setting.

Table 7 shows the learning results under the K-shot setting, from which we draw the following conclusions: (i) The key components we designed still play a non-negligible role in the "few-shot" experiments. For example, on HMDB-51/UCF-101 with $K = 2$, our model equipped with TP alone outperforms "Baseline" by 6.2%/9.8%, and the combination of TP and $\beta(\cdot)$ further improves accuracy by 3.5%/4.9%. Finally, there is a further improvement of 2.3%/2.9% through CN.Branch. (ii) Compared with state-of-the-art and representative methods, our model achieves higher performance gains with smaller K, as shown in Figure 8. We outperform some traditional single-modal methods by a large margin. Taking VideoSwin [15] as an example, our 2-shot model on HMDB-51 and UCF-101 is 40.7% and 29.4% higher, respectively. Such a large gap further verifies the effectiveness of transferring knowledge from VLMs to the video domain and the importance of improving the quality of pre-training data. Compared with the same VLM-based methods, our method is still better than [47], X-CLIP [33], and ActionCLIP [32] on HMDB-51 by 6.3%, 8.6%, and 6.8%, respectively, in the extreme case of $K = 2$, proving the powerful learning ability of our method even with extremely limited training data (i.e., about 3% of the videos in the training set). This also indicates that our Chinese text constraints based on CLIP have strengthened the semantic supervision of the model compared to the above methods, and compared to the full fine-tuning of X-CLIP [33] and ActionCLIP [32], only fine-tuning the proposed temporal prompt vectors and reparameterization encoder can retain the generalization ability of the CLIP pre-trained model to the greatest extent. This is the reason why our model can achieve advantages in "few-shot" environments. (iii) However, compared with the BIKE method, which is also based on VLMs, our model is inferior. We determined that the main reason is that the BIKE framework, based on the CLIP model, builds a complementary bridge between the visual and textual domains through bidirectional cross-modal knowledge mining, while we only strengthen video representations through an additional Chinese text branch without fully exploring the bidirectional knowledge between the two modalities. (iv) In summary, our method exhibits excellent transferability under data-scarce conditions and achieves leading "few-shot" performance on HMDB-51 and UCF-101 datasets.

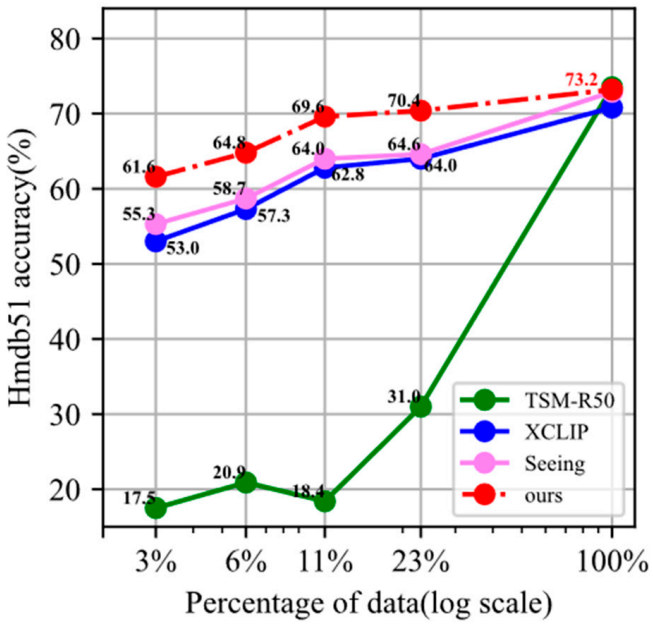


Figure 8. Performance comparison when data are extremely scarce.

Table 7. Few-shot training results on HMDB-51 and UCF-101 datasets under *K*-shot setting. The value range for *K* is {2, 4, 8, 16}. In the case of scarce data resources, we have significant performance advantages compared to methods other than BIKE.

Methods	Pretrain	TP	$\beta(\cdot)$	CN. Branch	HMDB-51				UCF-101			
					K							
					2	4	8	16	2	4	8	16
STM	ImageNet-1k	-	-	-	35.8	39.0	43.6	-	65.4	73.9	81.3	-
3D-ResNet-50					43.2	44.3	49.9	-	68.8	71.1	85.8	-
TSM-R50					17.5	20.9	18.4	31.0	25.3	47.0	64.4	61.0
TimeSformer	ImageNet-21k	-	-	-	19.6	40.6	49.4	55.4	48.5	75.6	83.7	89.4
Video Swin-B					20.9	41.3	47.9	56.1	53.3	74.1	85.8	88.7
X-Florence	FLD-900M	-	-	-	51.6	57.8	64.1	64.2	84.0	88.5	92.5	94.8
ActionCLIP	CLIP+ Kinetics-400	-	-	-	54.8	56.7	57.3	-	80.7	85.3	89.2	-
X-CLIP-B/16	CLIP-400M	-	-	-	53.0	57.3	62.8	64.0	76.4	83.4	88.3	91.4
Vita-B/16					39.9	44.5	54.0	57.0	70.1	79.3	83.7	90.0
BIKE-B/16					64.3	67.6	71.3	71.9	88.6	91.5	92.8	93.3
[47]					55.3	58.7	64.0	64.6	82.4	85.8	89.1	91.6
Baseline		-	-	-	49.6	52.8	57.3	60.3	65.1	74.8	77.6	80.2
Ours		✓			55.8	59.1	61.8	63.7	74.9	77.8	80.4	84.3
		✓	✓		59.3	62.4	66.0	67.9	79.8	82.1	84.8	88.2
		✓	✓	✓	61.6	64.8	69.6	70.4	82.7	86.0	89.5	92.1

4.4. Comparison with the State-of-the-Art

In this section, we compare the fully supervised performance under a "closed set" setting with the state-of-the-art video recognition models on three widely studied action recognition datasets. All experiments in this section are based on the CLIP [10] pre-trained model; the pre-trained weights are fixed for all layers. For HMDB-51 and UCF-101, the initial learning rate is set to 5×10^{-6} and the model is trained for 30 epochs; the batch size is 32. We test the accuracy of the split datasets using multi-view (i.e., 4 temporal clips $\times$ 3 spatial crops). In Table 8, we report the results on HMDB-51 and UCF-101 and compare our method with the previous methods conducted with different pre-training data, including K400, ImageNet, and web-scale vision–language pre-trained model (CLIP).

Table 8. Comparison between our method and state-of-the-art methods on HMDB-51 and UCF-101 datasets under fully supervised training settings. "Frozen" means freezing CLIP pre-trained parameters.

Methods	Pretrain Data	Modalities	Frozen	Top-1 (%)	
				UCF-101	HMDB-51
ARTNet	-	RGB	×	94.3	70.9
TSM				95.9	73.5
STM				96.2	72.2
MVFNet				96.6	75.7
TDN				97.4	76.4
R3D-50				92.0	66.0
NL-I3D				-	66.0

Table 8. *Cont.*

Methods	Pretrain Data	Modalities	Frozen	Top-1 (%)	
				UCF-101	HMDB-51
Methods with Kinetics pre-training					
STC	K400	RGB	×	95.8	72.6
ECO	K400			93.6	68.4
R(2+1)D-34	K400			96.8	74.5
FASTER32	K400			96.9	75.7
SlowOnly-8x8-R101	K400 + OmniSource			97.3	79.0
Methods with ImageNet pre-training					
I3D	ImageNet + K400	RGB	×	95.6	74.8
S3D	ImageNet + K400			96.8	75.9
LGD-3D	ImageNet + K600			97.0	75.7
Methods with large-scale image-language pre-training					
BIKE ViT-L	CLIP + K400	RGB	×	98.8	82.2
ViT-B/16 w/ST-Adapter	CLIP + K400		✓	96.4	77.7
VideoPrompt [17]	CLIP		✓	93.6	66.4
[47]	CLIP		✓	96.3	72.9
BIKE ViT-B	CLIP		×	96.6	73.3
XCLIP-B	CLIP		×	94.2	70.8
Vita ViT-B [48]	CLIP		✓	91.5	67.3
Ours ViT-B	CLIP		✓	96.5	73.8
Methods with additional modalities					
Two-Stream I3D	ImageNet + K400	RGB + Flow	×	98.0	80.7
Two-Stream LGD-3D	ImageNet + K600	RGB + Flow		98.2	80.5
PERF-Net	ImageNet + K700	RGB + Flow + Pose		98.6	83.2
SlowOnly-R101-RGB + I3D-Flow	OmniSource	RGB + Flow		98.6	83.8
SMART	ImageNet + K400	RGB + Flow		98.6	84.3

Compared with methods pre-trained on ImageNet, we find that our model only achieves comparable performance with I3D, and slightly lags behind other models. However, it is worth mentioning that they fine-tune based on the Kinetics dataset, although, our method fails to learn more good spatiotemporal representations from relevant fields to fit the model. The fully fine-tuned strategy requires them to save a large number of model parameters for each task, while our method not only has the advantage of a small number of adjustable parameters but can also save a lot of memory when facing increasingly large model backbones. Due to the limitation of computing resources, we believe there will be a significant performance improvement if our model is also pre-trained on Kinetics.

Compared with methods pre-trained on K400, our method outperforms STC and ECO based on RGB modality by 1.2%, 5.4% and 0.7%, 2.9% on HMDB-51 and UCF-101, respectively. We found that the performance gap mentioned above is partly due to the fact that traditional methods are mostly based on CNN architecture, in which the convolution operation can effectively capture local features of images; however, the ability to model global information over long distances is limited; on the contrary, our method is based on the ViT architecture, where the attention mechanism excels at modeling long-range temporal dependencies. However, compared with the SlowOnly-8×8-R101 method supported by OmniSource [49], we are clearly lagging behind, mainly due to the fact that the OmniSource framework can improve the model's performance on the given target dataset.

Our method is also competitive compared with the same pre-trained method using CLIP-400M. Compared with the [48] and [47] methods that also freeze the backbone network, the Top-1 accuracy of our method on HMDB-51 increases by +6.5%, and +0.9%, respectively. For the VideoPrompt [17] method, we have a significant performance gap of +7.4%. We believe that the +7.4% Top-1 gap is mainly due to the fact that VideoPrompt only adds a simple Transformer block after the frozen visual encoder and adds simple learnable prompts before and after text embedding to improve the quality of video embeddings, while our method not only adds new semantic constraints on the text but also re-parameterizes the temporal prompts through the encoder to enhance the temporal modeling ability of the model. Meanwhile, for X-CLIP [33] and BIKE ViT-B [50], which both fine-tune the visual backbone network, we not only have better performance but also have fewer trainable parameters, as shown in Figure 6a, which further proves that we can achieve a good transformation from large-scale image-based models to video models by only fine-tuning additional module parameters.

In addition, we also compared our method with methods based on multiple input modalities, and we found that our method is not advantageous or is even far inferior to them. However, this also inspired us to consider developing a CLIP-based model in a dual-stream or even multi-stream manner. Compared with optical flow-based methods, CLIP-based methods can avoid expensive optical flow calculation costs and improve speed.

Based on the above comparison, we can observe that, compared with popular video recognition models based on CNN and Transformer, our method maintains considerable competitiveness while also possessing the strong generalization ability demonstrated previously.

For the SSv1 dataset where action categories are less related to static backgrounds but closely associated with dynamic content, the model should be able to distinguish fine-grained behaviors in daily life, such as "digging something out of something", "letting something roll down a slanted surface", "moving something across a surface without it falling down", and other specific actions. To this end, different from the HMDB-51 and UCF-101 datasets, we used more powerful data augmentation techniques for the SSv1 dataset, including RandAugment, random erasing, and label smoothing. The initial learning rate was set to 5×10^{-4} and the model was trained for 40 epochs with a batch size of 16. Comparison of performance between our model and other state-of-the-art techniques on the SSv1 dataset is shown in Table 9.

Compared with methods based on the CNN architecture, our model has a slight advantage in terms of accuracy. Specifically, compared with methods based on ImageNet-1K pre-trained, our model has a performance gap of +7.7% at the highest and +0.8% at the lowest on the Top-1 accuracy. However, compared with methods based on ImageNet21K + K400 pre-trained, the performance of our model is slightly inferior. According to our speculation, this may be due to the large-scale and diverse data in the K400 dataset, which plays a vital role in the migration of the model to downstream tasks, mainly in enhancing the model's feature learning ability so that it can better capture temporal dynamic information in the samples.

Our method is also at a disadvantage among the methods based on Transformer architecture and pre-trained on cross-modal data. One reason for this may be that SSv1 is a "temporal-heavy" dataset that requires the model to understand the temporal changes in the video from the essence. Another reason is that the Chinese text branch we proposed can only constrain video representations semantically, while the proposed temporal prompt reparameterization encoder only adjusts task-specific parameters based on frozen CLIP parameters, without fully learning the temporal dependencies in the task. This reminds us that we need to fundamentally model temporal dynamics in our next work.

In summary, although our model has achieved relatively good results on the three video datasets mentioned above, there are still some potential limitations in practical applications, especially in processing data noise and video quality. During the training phase, although our model simulates video sources of various quality levels encountered in the

real world through data augmentation and other techniques to enhance its robustness under various conditions in real-world environments, we still need to focus on the actual application performance of the model in future research and applications when faced with extremely complex real-world situations in order to improve its accuracy and stability. Finally, our method is simple and feasible, and in future research, more powerful contrastive language–image pre-trained models may be utilized to enhance model performance. In addition, since temporal modeling on video data can be seen as a form of sequence modeling, it is highly likely that in the future we will reuse pre-trained weights from audio, 3D point clouds, and sensor models instead of image models.

Table 9. Comparison between our model and state-of-the-art methods on the SSv1 dataset under fully supervised training settings.

Methods	Pretrain Data	Architecture	Frozen	SSv1	
				Top-1	Top-5
Methods with ImageNet pre-training					
TANet-R50	ImageNet-1K	CNN	×	47.6	77.7
TSM				47.2	78.1
TEANet				48.9	-
SmallBig				50.0	79.8
STM				50.7	80.4
TEINet				51.0	-
AIA (TSM)				51.6	79.9
MSNet				52.1	82.3
TEA				52.3	81.9
SDA-TSM				52.8	81.3
CT-NET				53.4	81.7
TDN				53.9	82.1
TAdaConvNeXtV2-T				54.1	-
TAdaConvNeXtV2-S	ImageNet21K + K400			59.7	-
TAdaConvNeXtV2-B				60.7	-
Methods with large-scale image-language pre-training					
Ours-B/16	CLIP-400M	Transformer	✓	54.9	83.8
TAdaFormer-B/16			×	59.2	-
Side4Video-B/16			✓	60.7	86.0
UniFormerV2-B/16			×	56.8	84.2

5. Conclusions

In the field of video action recognition, using the spatial feature knowledge of large pre-trained image models to improve the spatiotemporal reasoning ability of the model is an important research direction. However, traditional methods often require full fine-tuning of the video model, which not only consumes computing resources but may also make it difficult to achieve optimal performance in specific application scenarios. Therefore, in order to address the above issue, in this work, we proposed a novel method that is extremely easy to implement without losing performance advantages for transferring the powerful spatial representation knowledge learned by large pre-trained image models to video action recognition where spatiotemporal reasoning capability is indispensable. Based on CLIP of frozen features, we designed a prompt vector for temporal modeling and further implemented model adaptation for specific domains through our proposed reparameterization encoder. The predefined Chinese label dictionary aimed to generate

video representations with semantic diversity through the introduced Chinese encoder. Extensive experiments in various learning scenarios demonstrate that our method achieves comparable or even better performance compared with prohibitive fully fine-tuned video models and existing state-of-the-art techniques.

Author Contributions: Conceptualization, L.D.; methodology, L.D. and J.T.; data curation, L.D. and F.L.; formal analysis, L.D. and F.L.; software, J.T.; validation, J.T. and L.D.; writing—review and editing, J.T.; investigation, L.D. and F.L.; supervision, L.D. and F.L.; writing—original draft preparation, J.T. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Natural Science Foundation of Henan (Grant No. 242300421220) and the Henan Provincial Science and Technology Research Project under Grants 242102211007, 242102211020, 232102211006, 232102210044, and 232102211017.

Data Availability Statement: The data presented in this study are openly available in [mmaction2] at [10.1109/ICCV.2011.6126543] and [mmaction2] at [arXiv:1212.0402v1] and [mmaction2] at [arXiv:1706.04261v2].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sahoo, J.P.; Prakash, A.J.; Plawiak, P.; Samantray, S. Real-time hand gesture recognition using fine-tuned convolutional neural network. *Sensors* **2022**, *22*, 706. [CrossRef] [PubMed]
2. Jiang, Q.; Li, G.; Yu, J.; Li, X. A model based method of pedestrian abnormal behavior detection in traffic scene. In Proceedings of the 2015 IEEE First International Smart Cities Conference (ISC2), Guadalajara, Mexico, 25–28 October 2015.
3. Lentzas, A.; Vrakas, D. Non-intrusive human activity recognition and abnormal behavior detection on elderly people: A review. *Artif. Intell. Rev.* **2020**, *53*, 1975–2021. [CrossRef]
4. Tang, Z.; Gu, R.; Hwang, J.N. Joint multi-view people tracking and pose estimation for 3D scene reconstruction. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018.
5. Carreira, J.; Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
6. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
7. Tran, D.; Wang, H.; Torresani, L.; Feiszli, M. Video classification with channel-separated convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
8. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
9. Selva, J.; Johansen, A.S.; Escalera, S.; Nasrollahi, K.; Moeslund, T.B.; Clapés, A. Video transformers: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12922–12943. [CrossRef]
10. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning (PMLR), Virtual, 18–24 July 2021.
11. Yuan, L.; Chen, D.; Chen, Y.L.; Codella, N.; Dai, X.; Gao, J.; Hu, H.; Huang, X.; Li, B.; Li, C.; et al. Florence: A new foundation model for computer vision. *arXiv* **2021**, arXiv:2111.11432.
12. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Learning to prompt for vision-language models. *Int. J. Comput. Vis.* **2022**, *130*, 2337–2348. [CrossRef]
13. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
14. Xu, H.; Ghosh, G.; Huang, P.Y.; Okhonko, D.; Aghajanyan, A.; Metze, F.; Zettlemoyer, L.; Feichtenhofer, C. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv* **2021**, arXiv:2109.14084.
15. Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; Hu, H. Video swin transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
16. Lester, B.; Al-Rfou, R.; Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv* **2021**, arXiv:2104.08691.
17. Ju, C.; Han, T.; Zheng, K.; Zhang, Y.; Xie, W. Prompting Visual-Language Models for Efficient Video Understanding. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
18. Lin, Z.; Geng, S.; Zhang, R.; Gao, P.; De Melo, G.; Wang, X.; Dai, J.; Qiao, Y.; Li, H. Frozen clip models are efficient video learners. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
19. Pan, J.; Lin, Z.; Zhu, X.; Shao, J.; Li, H. St-adapter: Parameter-efficient image-to-video transfer learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 26462–26477.

20. Jia, M.; Tang, L.; Chen, B.C.; Cardie, C.; Belongie, S.; Hariharan, B.; Lim, S.N. Visual prompt tuning. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
21. Bahng, H.; Jahanian, A.; Sankaranarayanan, S.; Isola, P. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv* **2022**, arXiv:2203.17274.
22. Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16664–16678.
23. Jie, S.; Deng, Z.H. Convolutional bypasses are better vision transformer adapters. *arXiv* **2022**, arXiv:2207.07039.
24. Gao, Y.; Shi, X.; Zhu, Y.; Wang, H.; Tang, Z.; Zhou, X.; Li, M.; Metaxas, D.N. Visual prompt tuning for test-time domain adaptation. *arXiv* **2022**, arXiv:2210.04831.
25. Li, X.L.; Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv* **2021**, arXiv:2101.00190.
26. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. GPT understands, too. *AI Open*, 2023; *in press*.
27. Wang, X.; Zhu, L.; Wang, H.; Yang, Y. Interactive prototype learning for egocentric action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
28. Stroud, J.; Ross, D.; Sun, C.; Deng, J.; Sukthankar, R. D3d: Distilled 3d networks for video action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020.
29. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
30. Wang, L.; Tong, Z.; Ji, B.; Wu, G. Tdn: Temporal difference networks for efficient action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
31. Yan, S.; Xiong, X.; Arnab, A.; Lu, Z.; Zhang, M.; Sun, C.; Schmid, C. Multiview transformers for video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
32. Wang, M.; Xing, J.; Liu, Y. Actionclip: A new paradigm for video action recognition. *arXiv* **2021**, arXiv:2109.08472.
33. Ni, B.; Peng, H.; Chen, M.; Zhang, S.; Meng, G.; Fu, J.; Xiang, S.; Ling, H. Expanding language-image pretrained models for general video recognition. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022.
34. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
35. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
36. Rao, Y.; Zhao, W.; Chen, G.; Tang, Y.; Zhu, Z.; Huang, G.; Zhou, J.; Lu, J. Denseclip: Language-guided dense prediction with context-aware prompting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
37. Du, Y.; Wei, F.; Zhang, Z.; Shi, M.; Gao, Y.; Li, G. Learning to prompt for open-vocabulary object detection with vision-language model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
38. Zhang, R.; Guo, Z.; Zhang, W.; Li, K.; Miao, X.; Cui, B.; Qiao, Y.; Gao, P.; Li, H. Pointclip: Point cloud understanding by clip. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
39. Lei, J.; Li, L.; Zhou, L.; Gan, Z.; Berg, T.L.; Bansal, M.; Liu, J. Less is more: Clipbert for video-and-language learning via sparse sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
40. He, J.; Zhou, C.; Ma, X.; Berg-Kirkpatrick, T.; Neubig, G. Towards a unified view of parameter-efficient transfer learning. *arXiv* **2021**, arXiv:2110.04366.
41. Guo, D.; Rush, A.M.; Kim, Y. Parameter-efficient transfer learning with diff pruning. *arXiv* **2020**, arXiv:2012.07463.
42. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
43. Yang, A.; Pan, J.; Lin, J.; Men, R.; Zhang, Y.; Zhou, J.; Zhou, C. Chinese clip: Contrastive vision-language pretraining in chinese. *arXiv* **2022**, arXiv:2211.01335.
44. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
45. Huang, Z.; Zhang, S.; Pan, L.; Qing, Z.; Tang, M.; Liu, Z.; Ang Jr, M.H. Tada! temporally-adaptive convolutions for video understanding. *arXiv* **2021**, arXiv:2110.06178.
46. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
47. Wang, Q.; Du, J.; Yan, K.; Ding, S. Seeing in flowing: Adapting clip for action recognition with motion prompts learning. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023.
48. Wasim, S.T.; Naseer, M.; Khan, S.; Khan, F.S.; Shah, M. Vita-clip: Video and text adaptive clip via multimodal prompting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.

49. Duan, H.; Zhao, Y.; Xiong, Y.; Liu, W.; Lin, D. Omni-sourced webly-supervised learning for video recognition. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
50. Wu, W.; Wang, X.; Luo, H.; Wang, J.; Yang, Y.; Ouyang, W. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition, Vancouver, BC, Canada, 17–24 June 2023.