PromptLearner-CLIP: Contrastive Multi-Modal Action Representation Learning with Context Optimization

Zhenxing Zheng[1,2,3], Gaoyun An[2,3,⋆], Shan Cao[2,3],
Zhaoqilin Yang[2,3], and Qiuqi Ruan[2,3]

[1] School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[2] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[3] Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China
zhxzheng@xupt.edu.cn {gyan,18112001,19112010,qqruan}@bjtu.edu.cn

Abstract. An action contains rich multi-modal information, and current methods generally map the action class to a digital number as supervised information to train models. However, numerical labels cannot describe the semantic content contained in the action. This paper proposes PromptLearner-CLIP for action recognition, where the text pathway uses PromptLearner to automatically learn the text content of prompt as the input and calculates the semantic features of actions, and the vision pathway takes video data as the input to learn the visual features of actions. To strengthen the interaction between features of different modalities, this paper proposes a multi-modal information interaction module that utilizes Graph Neural Network(GNN) to process both the semantic features of text content and the visual features of a video. In addition, the single-modal video classification problem is transformed into a multi-modal video-text matching problem. Multi-modal contrastive learning is used to disclose the feature distance of the same but different modalities samples. The experimental results showed that PromptLearner-CLIP could utilize the textual semantic information to significantly improve the performance of various single-modal backbone networks on action recognition and achieved top-tier results on Kinetics400, UCF101, and HMDB51 datasets. Code is available at https://github.com/ZhenxingZheng/PromptLearner.

1 Introduction

With the development of mobile devices and communication networks, video has become the main carrier of information. It is of great practical significance to understand and analyze human actions in the video. As an essential branch of video understanding, action recognition aims to analyze and recognize human actions in videos by analyzing video data and using specific algorithms.

[⋆] Corresponding author

Different from image processing tasks, action recognition needs to analyze not only the appearance information but also the semantics of an action. How to effectively encode the feature of an action remains a fundamental problem to be solved. An action contains rich multi-modal information, and current methods generally map the action class to a digital number as supervised information to train models. However, numerical labels cannot describe the semantic content contained in the action. The sample **playing tennis** on Youtube provides the corresponding text description of **A 12-year-old boy playing tennis**, which not only describes the class of the action but also includes the action subject. Therefore, the text provides rich semantic information and the visual feature can be enhanced by relevant text content.

Although some samples on Youtube are accompanied by detailed text descriptions, most samples contain a lot of information unrelated to the video content. Recently, in the field of natural language processing, researchers proposed a new paradigm: "pretrain, prompt, predict", where according to the downstream task, a template is designed such that the model can fit the task of pre-training when predicting. Based on this, CLIP [29] constructed the text input by designing a variety of natural language description templates and filled the image label text into the blank positions of templates. The experiments showed that different prompts have an important impact on the model, and subtle differences may lead to changes in performance. CoOp [52] used continuous representations to represent prompt whose parameters are optimized in an end-to-end fashion.

Based on the above analyses, this paper proposes a multi-modal semantic-guided network PromptLearner-CLIP for action recognition. In the training phase, the text labels of the samples in a batch are filled in the prompt template and then processed by the text encoder to extract text features. At the same time, the visual features of the videos in the batch are extracted by the visual encoder. Finally, the similarities of visual features and text features are computed, resulting in the similarity matrix that is used for optimization. In the inference phase, all labels are filled in the prompt template and the feature similarity scores between each video in the test dataset and all prompts are computed. The label with the highest similarity score is assigned to the video. Our contributions are summarized as follows: (1) PromptLearner is used to learn the text content of prompt as the input to the text pathway and its parameters are optimized together with the backbone network; (2) GNN is used to process both the semantic features of text content and the visual features of the video and strengthen the interaction between semantic features and visual features; (3) Finally, the Kullback-Leibler(KL) loss and supervised contrastive loss are used to disclose the feature distance of the same but different modalities samples.

2 Related Work

Single-Modal Action Recognition. C3D [33] stacked 3D convolutional layers to learn spatial-temporal features. ARTNet [37] designed appearance and relation branches to perform spatial modeling and relation modeling in a parallel way. R(2+1)D [34] decomposed the 3D convolution kernel into a 2D spatial convolution kernel and a 1D temporal convolution kernel. Because action recognition needs to process multiple frames of a video, it has large computational complexity. Based on the fact that adjacent frames of a video have redundant information, TSN [39] proposed a sparse sampling strategy and a feature aggregation module to model long-term temporal relationships of an action. AdaScan [15] pooled the video frames containing important information and discarded the video frames with less information. To complete effective temporal modeling for actions, TSM [24] shifted the feature map by a position along the temporal dimension, so that the convolutional feature map of the current frame obtains the information of adjacent frames.

Cross-Modal Action Recognition. PoTion [5] encoded the displacement of key points of the human body on a color image that was then processed by CNN to obtain action features containing pose motion information. Multi-stream network [43] used three streams to process an RGB image, multiple optical flow images, and spectrograms to model appearance features, short-term motion features, and sound features of actions respectively. In addition to sound information, the text as a rich expression can describe the semantic content of a video. CLIPBERT [19] fused the feature of each video clip and the feature of text to model the multi-modal feature by Transformer [36]. ActBERT [53] learned joint video-text feature representations to capture global and local visual cues from each pair of the video clip and text description.

Contrastive Learning. Contrastive learning as an unsupervised representation learning method has been successfully applied in the field of computer vision. MoCo [12] built a dynamic dictionary with a queue composed of previous sample features and set an instance discrimination task for contrastive unsupervised learning. VideoMoCo [28] built a generator to mask partial video frames and used a discriminator to distinguish the full video sequence features from the masked video sequence features. Inspired by the mask prediction task, MaskCo [50] masked a specific region of an enhanced image while keeping the other enhanced image unchanged, and then calculated the region-level feature contrastive loss of two images to implement the contrastive mask prediction task for visual representation learning. In the training of a network, a batch of samples may contain multiple samples belonging to the same class. SupCon [17] incorporated the label information into contrastive loss, which considers multiple positive samples of the same class for each anchor point so that the features from the same class are closer than the features of different classes.

3 Method

This paper proposes a multi-modal semantic-guided action recognition network PromptLearner-CLIP for action recognition, as shown in Fig. 1. The vision pathway uses ViT [7] as the backbone network to process video frames and obtains frame features and video features. The text pathway uses Transformer to process text content and obtains word features and sentence features. To construct valid text input, PromptLearner is used to automatically learn text content as the input to the text pathway. After extracting text features and visual features, a multi-modal information interaction module is used to interact with text features and visual features. Finally, multi-modal contrastive learning is used to disclose the feature distance of the videos belonging to the same class.

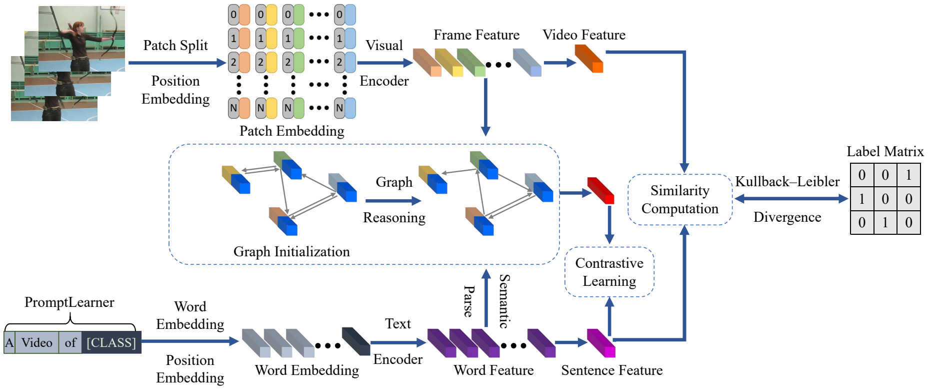

Fig. 1. Illustration of PromptLearner-CLIP consisting of vision feature extraction, text feature extraction, multi-modal information interaction, and contrastive learning.

3.1 Text Pathway

Multi-modal learning aims to use specific algorithms to learn the complementarity and eliminate the redundancy between different modalities. However, most of the datasets for action recognition only provide action classes without corresponding text descriptions. Recently, a new paradigm of prompt has been proposed in the field of natural language processing. By setting different fill-in-the-blank templates, the downstream task is adjusted to the form similar to the pre-training task, which can effectively solve the downstream task. PromptLearner proposed in this paper uses a text template to process action label text by expanding the label text into the sentence with certain semantic content as text descriptions of a video, which is used as the input to the text pathway to extract semantic information. PromptLearner uses a continuous vector to represent the content of text, and its parameters are updated together with the backbone network in an end-to-end fashion, represented as follows:

$$t^i = [V_1^i][V_2^i][V_3^i]...[V_M^i][CLASS^i], \tag{1}$$

where $[V_m](m \in [1, M])$ represents the context token, $[CLASS^i]$ is the i-th label text of the action, t^i is the learnable text content, $i \in [1, I]$, and M and I represent the number of context tokens and action classes respectively. Class-specific context token is used in this paper.

After the learnable prompt is constructed, Transformer is used to process the text content. Transformer adopts an encoder-decoder structure and the text pathway only uses the Transformer encoder to extract the features of text content. The encoder consists of multiple encoding layers consisting of the self-attention layer, LayerNorm layer, and feed-forward layer. The structure of an encoder is shown in Fig. 2.

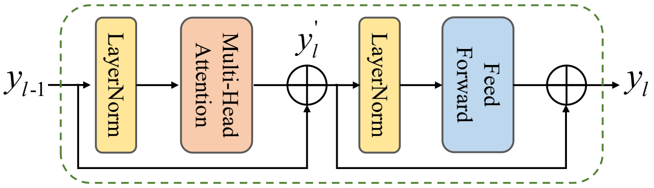

Fig. 2. Structure of Transformer encoder consisting of the self-attention layer, LayerNorm layer, and feed-forward layer.

The overall process of the Transformer encoder extracting text features is represented as follows:

$$y_0 = [V_1 + PE_1, \ldots, V_M + PE_M, V_{class} + PE_{class}], \tag{2}$$

$$q_l = k_l = v_l = \text{LayerNorm}(y_{l-1}), \tag{3}$$

$$y_l' = \text{MSA}(q_l, k_l, v_l) + y_{l-1}, \tag{4}$$

$$y_l = \text{FFN}(\text{LayerNorm}(y_l')) + y_l', \quad l = 1, \ldots, L \tag{5}$$

$$[s_1, s_2, \ldots, s_M, s_{class}] = y_L, \tag{6}$$

where l denotes the l-th encode layer, V_m and PE_m denote the m-th context token and positional embedding, MSA denotes the multi-head self-attention layer, FFN denotes the feed-forward layer, and L is the number of Transformer encoder layers. Transformer uses the self-attention layer to effectively capture dependencies between features at any location and learn text features. After the text content is processed by the Transformer encoder, the word features $S = \{s_m\}_{m=1}^{M+1}$ and sentence features s_0 are obtained.

3.2 Vision Pathway

The original input to Transformer is a sequence of tokens. To satisfy the requirements of Transformer, ViT first pre-processes the frame to obtain the input token sequence. Given a frame $x \in R^{H \times W \times C}$, ViT first splits the frame into many non-overlapped patches of the same size, and then these frame patches are flattened into the 1D vectors composed of pixel values, denoted as $x_p \in R^{N \times (P^2 \cdot C)}$:

$$x_p = [x_p^1, x_p^2, \ldots, x_p^N], \tag{7}$$

where H and W denote the height and width of the frame respectively, C is the number of channels, (P, P) is the resolution of a patch, x_p^n represents the flatten vector from the n-th patch, and $N = H \cdot W/P^2$ is the number of split patches. Through linear projection $\mathbf{E}$ consisting of fully-connected layers, the vector consisting of pixels is transformed to the patch embedding feature with the dimension of d_{model}. At the start of the patch sequence, we prepend a learnable embedding z_{cls} and its state at the output layer denotes the frame feature. A learnable 1D position embedding is used to retrain the position information of each patch and is added to the patch embedding feature, as shown as follows:

$$z_0 = [z_{cls}, \mathbf{E}x_p^1, \mathbf{E}x_p^2, \ldots, \mathbf{E}x_p^N] + \mathbf{p}, \tag{8}$$

The overall process of ViT extracting the frame feature is summarized as follows:

$$q_l = k_l = v_l = \mathrm{LayerNorm}(z_{l-1}), \tag{9}$$

$$z'_l = \mathrm{MSA}(q_l, k_l, v_l) + z_{l-1}, \tag{10}$$

$$z_l = \mathrm{FFN}(\mathrm{LayerNorm}(z'_l)) + z'_l. \quad l = 1, \ldots, L \tag{11}$$

Finally, Transformer is used to process the video frame by frame and we obtained the sequence of frame features $V = \{v_k\}_{k=1}^K$. These features are averaged as the video-level feature v_0.

3.3 Multi-Modal Information Interaction Module

After extracting vision features and text features, the model obtains frame-level features $V = \{v_k\}_{k=1}^K$, the video-level feature v_0, word-level features $S = \{s_m\}_{m=1}^{M+1}$, and the sentence-level feature s_0. Inspired by Dynamic Graph Attention Network [47], the multi-modal information interaction module represents the sentence as multiple soft distributions over words and parses the language structure of the sentence gradually. Firstly, the sentence-level feature s_0 is linearly projected to the question feature q^t at the t-th step and is concatenated with the results of the previous step, resulting in u^t:

$$q^t = W^t \times s_0 + b^t, \tag{12}$$

$$u^t = [q^t, y^{t-1}], \tag{13}$$

where W^t and b^t denote the learnable parameters, y^{t-1} denotes the results at the $(t-1)$-th step. Then, the semantic parsing module computes the similarity between u^t and each word-level feature to predict the visual reasoning processing, obtaining the soft distribution over all words $R^t = \{r_m^t\}_{m=1}^{M+1}$:

$$s^t = \delta(W_u \times u^t + b_u), \tag{14}$$

$$a_m^t = W_{s2} \times [\tanh(W_{s0} \times s^t + W_{s1} \times s_m)], \tag{15}$$

$$r_m^t = \frac{\exp(a_m^t)}{\sum\limits_{m=1}^{M+1} \exp(a_m^t)}, \tag{16}$$

where W_u, b_u, W_{s0}, W_{s1}, and W_{s2} are parameter matrices and are shared at different visual reasoning steps, and δ denotes ReLU. Finally, the output at the t-step is calculated as follows:

$$y^t = \sum_{m=1}^{M+1} r_m^t \cdot s_m. \tag{17}$$

Then, the semantic parsing feature y^t and the frame-level features $V = \{v_k\}_{k=1}^{K}$ are jointly fed into GNN for multi-modal feature interaction. To effectively embed textual semantic information into visual features, a question-guided graph attention mechanism is used to dynamically assign higher weights to frame features related to textual content. In this paper, and all frame features are concatenated with the output of the semantic parsing module as the vertices of the graph. The edges connecting the vertices represent the relationship between frames, and the vertex features are represented as follows:

$$v_k^{'} = [v_k, y^t], \quad \text{for} \quad k = 1, ..., K. \tag{18}$$

Next, the self-attention layer is used to calculate the correlation between the feature of any vertex in the graph and the features of all neighboring vertices, the neighboring vertex information is aggregated to update the vertex feature. Feature correlation is calculated as follows:

$$\alpha_{ij}^v = (W_q \times v_i^{'}) \times (W_k \times v_j^{'})^T, \tag{19}$$

where W_q and W_k are parameter matrices used for projecting the vertex feature into the feature subspace in which the correlations between all other vertex features and the i-th vertex feature are computed. The correlation α^v is normalized by the softmax function as a weight to aggregate the information of other vertices:

$$\alpha_{ij} = \frac{\exp(\alpha_{ij}^v)}{\sum\limits_{j=1, j\neq i}^{K} \exp(\alpha_{ij}^v)}, \tag{20}$$

$$v_i^* = \delta(v_i^{'} + \sum_{j, j\neq i} \alpha_{ij} \cdot v_j^{'}), \tag{21}$$

where α_{ij} denotes the weight between the i-th and j-th vertex features.

Finally, the multi-modal fusion method BUTD [1] is used to obtain the multi-modal representation:

$$\boldsymbol{J} = f(v^*, y^t; W_{fuse}), \tag{22}$$

where W_{fuse} denotes the parameter of the fusion method, and $\boldsymbol{J}$ is the resulted multi-modal feature.

3.4 Cross-Modal Contrastive Learning

Cross-modal contrastive learning plays an important role in image retrieval by learning a shared feature space for image-text matching. Therefore, the matching loss between the multi-modal feature similarity matrix and the label matrix in ActionCLIP [40] is used to pull the pairwise text and visual features close to each other:

$$\mathcal{L}^{KL} = \frac{1}{2}\mathbb{E}_{(\boldsymbol{s},\boldsymbol{v})\sim\mathcal{D}}[\mathrm{KL}(\boldsymbol{p}^{\boldsymbol{s2v}}(\boldsymbol{s}), \boldsymbol{q}^{\boldsymbol{s2v}}(\boldsymbol{s})) + \mathrm{KL}(\boldsymbol{p}^{\boldsymbol{v2s}}(\boldsymbol{v}), \boldsymbol{q}^{\boldsymbol{v2s}}(\boldsymbol{v}))], \tag{23}$$

where $\boldsymbol{q}^{\boldsymbol{s2v}}(\boldsymbol{s})$ and $\boldsymbol{q}^{\boldsymbol{v2s}}(\boldsymbol{v})$ are label matrices where the position of pairwise video and text is set to 1 and other positions are set to 0. $\boldsymbol{p}^{\boldsymbol{s2v}}(\boldsymbol{s})$ and $\boldsymbol{p}^{\boldsymbol{v2s}}(\boldsymbol{v})$ are multi-modal feature similarity matrices where cosine distance is used to measure the feature similarity.

Although KL loss can disclose the difference between the multi-modal feature similarity matrix and the label matrix, there may be multiple positive sample pairs belonging to the same class in a batch of samples. If the label information is included in contrastive learning, the feature encoder will produce the features at a closer distance. The supervised contrastive learning is calculated as follows:

$$\mathcal{L}^{sup} = \sum_{i\in\mathcal{D}}\mathcal{L}_i^{sup} = \sum_{i\in\mathcal{D}}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(\boldsymbol{J}_i \times \boldsymbol{s}_0^p/\tau)}{\sum\limits_{a\in A(i)}\exp(\boldsymbol{J}_i \times \boldsymbol{s}_0^a/\tau)}, \tag{24}$$

where $P(i) \equiv \{p \in P(i) : y_p = y_i\}$ is the set of indices of all positives to the i-th sample in a batch, $|P(i)|$ is its cardinality, $A(i) \equiv I \setminus \{i\}$, and $\mathcal{D}$ denotes a batch of samples. The overall loss is represented as follows:

$$\mathcal{L} = \mathcal{L}^{KL} + \mathcal{L}^{sup}. \tag{25}$$

4 Experiments

4.1 Datasets

The training set of Kinetics400 [16] has 240K videos and the validation set has 20K videos. Kinetics400 is divided into 400 categories, each of which contains at least 400 samples. Each sample is obtained by cropping Youtube videos and lasts about 10 seconds.

Mini-Kinetics-200 [44] is a subset of the Kinetics400 dataset and contains 200 categories. There are 400 samples and 25 samples for each category in the training set and the validation set respectively.

The UCF101 [31] dataset contains 13 320 videos with a total of 101 action categories. The HMDB51 [18] dataset contains 51 categories of daily actions with a total of 6 766 videos. UCF101 and HMDB51 datasets provide three splits of training sets and testing sets, and researchers need to compare the average accuracy of the three splits to verify the effectiveness of the method.

4.2 Implementation Details

The text encoder adopts Transformer with 12 layers, where the self-attention layer contains 8 heads and the number of neurons in the hidden layer is set to 512. For the visual feature encoder, this paper uses ViT that also has a 12-layer Transformer. Two types of feature encoders are initialized from CLIP [29]. The number of learnable context tokens of PromptLearner is set to 16. The initial template is "a video of action X", where X represents the label text of the action and the context vector is randomly initialized with mean 0 and variance 0.02. For the multi-modal information interaction module, the number of neurons in the hidden layer of GNN is set to 512, and the dimension of the output composite feature is set to 512.

The AdamW optimizer is used to optimize the model's parameters, where the initial learning rates of the encoders and the remaining module parameters are set to 5e-6 and 5e-5 respectively, and the weight decay is set to 0.2. The batch size is set to 64, and the total training epoch is set to 50. The first 5 epochs use the warm-up strategy and the remaining 45 epochs use the half-cosine decay strategy. RandAugment is used to crop the region with 224 × 224 size from each frame. This paper adopts a sampling method to randomly sample 8 or 16 frames from a video. During the testing, 10 groups of frame sequences were randomly sampled from each video, and the average of the 10 groups of similarity scores is calculated to predict the action category. After the model is trained on Kinetics400, PromptLearner-CLIP is transferred to UCF101 and HMDB51 datasets, keeping the training and testing strategies unchanged.

4.3 Ablation Study

The loss function mainly consists of three parts, the video-text supervised contrastive loss, the video-text KL loss, and the text-video KL loss. The numbers in the first column of Table 1 represent the coefficients of corresponding loss functions, and the second and third columns report Top-1 and Top-5 accuracies on mini-Kinetics-200 respectively.

Table 1. Analysis of loss functions on mini-Kinetics-200 (%)

contrastive loss:video-text loss:text-video loss	Top-1	Top-5
1:0:1	67.15	88.89
1:1:0	84.70	97.29
0:1:1	85.10	97.39
1:1:1	85.34	97.15

The experimental results in Table 1 show that when three loss functions are used to optimize parameters, PromptLearner-CLIP achieves the best experimental results on mini-Kinetics-200, and the Top-1 accuracy is 85.34%. When the video-text KL loss is removed, the Top-1 accuracy drops to 67.15%. At the same time, it can be seen from the third row of the table that when the text-video KL loss is removed, the Top-1 accuracy of the model drops to 84.70%. Finally, observing the experimental results in the fourth row when removing the multi-modal supervised contrastive loss, the model has a little drop in the Top-1 accuracy, which confirms that the multi-modal supervised contrastive loss can further increase the similarity of samples belonging to the same class and improve the performance of the model.

Table 2 builds multiple models to study the influence of different semantic information on the multi-modal information interaction module. The average feature in the first column means that the average vector of word features encoded by Transformer is used as the input, and the Transformer feature means that the output at [EOS] position of the highest layer is used as the input. The models from the fourth row to the seventh row represent the text semantic parsing features with different semantic parsing steps.

Table 2. Analysis of the multi-modal information interaction module on mini-Kinetics-200 (%)

Textual Information	Top-1	Top-5
average feature	85.16	97.11
Transformer feature	85.04	97.49
one-step semantic parse	84.74	97.27
two-step semantic parse	85.02	97.17
three-step semantic parse	85.26	97.05
four-step semantic parse	85.06	97.19

Table 2 shows that when the average feature and Transformer feature are used as the inputs, the model achieves similar performance, which demonstrates that both features can effectively represent the input without parsing text semantics. The experimental results in the fourth row to the seventh row processed by different semantic parsing steps show that the accuracy of the model is gradually increased, and the model with three parsing steps achieves the highest value, rising from 84.74% to 85.26%, which indicates that parsing the text content can provide more detailed semantic information to guide the learning process of visual features.

Table 3 conducts ablation analyses of PromptLearner-CLIP on mini-Kinetics-200. For a fair comparison, a baseline model was set up to use ViT to process visual input. ActionCLIP uses the single-modal model ViT as the backbone network for visual feature extraction and builds a multi-modal learning framework to process visual data and text data simultaneously. The fourth, fifth, and sixth rows of Table 3 represent the results removing the multi-modal information interaction module, PromptLearner initialization, and video-text supervised contrastive loss respectively.

Table 3. Ablation study of PromptLearner-CLIP on mini-Kinetics-200 (%)

Interaction Module	Prompt Initializaiton	Contrastive Learning	Top-1	Top-5
-	-	-	83.70	96.53
-	-	-	84.02	96.85
-	✓	✓	84.78	97.41
✓	-	✓	84.82	97.27
✓	✓	-	85.10	97.39
✓	✓	✓	85.26	97.05

When textual information is incorporated, ActionCLIP improves the classification results of ViT on mini-Kinetics-200, which demonstrates that text content can provide semantic clues for action recognition. From the fourth to seventh rows in the table, when the main modules in PromptLearner-CLIP are removed one by one, the performance decreases to different degrees, indicating that each module in the model has a certain contribution to improving the performance. When the model removes the multi-modal information interaction module, the accuracy in Top-1 drops to 84.78%. Compared with the experimental results of the fifth and sixth rows, the performance drops the most.

4.4 Pathway Finetune

PromptLearner-CLIP contains text and vision pathways to extract features of different modalities. Table 4 summarizes the results of fine-tuning the backbone network parameters of different pathways on mini-Kinetics-200. Table 4 shows that the model finetuning two pathways(the fifth row) is significantly higher than the model freezing the parameters of two pathways(the second row) in the Top-1 accuracy, and the Top-1 and Top-5 accuracies are increased by 3.71% and 0.86% respectively, which demonstrates that it is still necessary to finetune the model parameters on the target dataset and learn the dataset-specific features.

Table 4. Analysis of finetuning different pathways on mini-Kinetics-200 (%)

Text Pathway	Vision Pathway	Top-1	Top-5
-	-	81.55	96.19
✓	-	81.73	95.93
-	✓	85.04	97.29
✓	✓	85.26	97.05

4.5 Different Backbone Networks

In this section, the visual backbone networks have completed the training on Kinetics400 and the parameters of them are frozen. Table 5 shows the accuracies of PromptLearner-CLIP using three visual backbone networks on Kinetics400. In the comparison of each group of the same backbone network, PromptLearner-CLIP achieves performance improvement on ActionCLIP. The experimental results in the third and fifth rows of the table show that reducing the size of frame patches will bring more computation, but the model will learn more detailed relationships between frame regions and discriminative features. The experimental results in the fifth and seventh rows show that more video frames can help the model to obtain more complete action information and improve the accuracy of the model.

Table 5. Analysis of different visual backbone networks on Kinetics400 (%)

Model	Top-1	Top-5
ActionCLIP(ViT-32-8f)	77.49	93.88
PromptLearner-CLIP(ViT-32-8f)	77.92	94.51
ActionCLIP(ViT-16-8f)	80.32	95.41
PromptLearner-CLIP(ViT-16-8f)	80.86	95.61
ActionCLIP(ViT-16-16f)	81.12	95.73
PromptLearner-CLIP(ViT-16-16f)	81.60	95.86

4.6 Comparison with State-of-the-Art Methods

Finally, PromptLearner-CLIP and current state-of-the-art methods are compared on the Kinetics400 dataset. Table 6 summarizes Top-1 and Top-5 accuracies on Kinetics400 of different methods.

First, the classification accuracies of Transformer-based methods in the table, such as MViT-B [8], TimeSformer-L [42], and ViT-B-VTN [27] on Kinetics400 are higher than 2D CNN-based [23] and 3D CNN-based [10] methods. PromptLearner-CLIP uses the same backbone network and achieves higher experimental results, although the number of input frames to the model is less than these three models. ViViT-L [2] uses a deeper ViT as the backbone network and achieves 80.6% Top-1 accuracy on Kinetics400, which is lower than our model by 1.0%. When ViViT-L initializes ViT-B/16 parameters with the model pre-trained on the large-scale dataset JFT, ViViT-L achieves the best experimental results in the table. Deeper models, more video frames, larger image resolutions, and larger pre-training datasets will stimulate the potential of the model to achieve a higher action recognition accuracy.

4.7 Finetune on Small Datasets

Finally, the PromptLearner-CLIP pre-trained on Kinetics400 is transferred to HMDB51 and UCF101, the results are shown in Table 7. MSM-ResNets [54] takes RGB images, optical flow images, and action saliency images as inputs,

Table 6. Comparison with state-of-the-art methods on Kinetics400 (%)

Model	Source	Top-1	Top-5
TEA-ResNet50 [23]	CVPR2020	76.1	92.5
TEINet [25]	AAAI2020	76.2	92.5
SmallBigNet [22]	CVPR2020	77.4	93.3
SVT [30]	CVPR2022	78.1	-
TPN-R101 [46]	CVPR2020	78.9	93.9
TANet [26]	ICCV2021	79.3	94.1
TDN [38]	CVPR2021	79.4	93.9
SlowFast [11]	ICCV2019	79.8	93.9
ViT-B-VTN [27]	ICCV2021	79.8	94.2
X3D-XXL [10]	CVPR2020	80.4	94.7
TokenShift [48]	ACM2021	80.4	94.5
BEVT [41]	CVPR2022	80.6	-
ViViT-L [2]	ICCV2021	80.6	94.7
TimeSformer-L [3]	ICML2021	80.7	94.7
MViT-B [8]	ICCV2021	81.2	95.1
DirecFormer[35]	CVPR2022	82.8	94.9
ViViT-L(JFT) [2]	ICCV2021	**82.8**	95.3
PromptLearner-CLIP	-	81.6	**95.9**

which obtains 66.7% on HMDB51 and 93.5% on UCF101. Since MSM-ResNets is not pre-trained on large-scale video datasets, the performance of the model is significantly lower than the current state-of-the-art methods. PoTion [5] extracts the motion information of human pose from the video to learn pose motion features. Two-stream I3D [4] extracts appearance features and action features from RGB images and optical flow images respectively. When the spatial-temporal features of I3D are fused with the PoTion features, the accuracies are improved by 0.7% on HMDB51 and 0.3% on UCF101. However, I3D+PoTion is lower than PromptLearner-CLIP+I3D(Flow) on both datasets, revealing that the multi-modal learning framework proposed in this paper can effectively utilize the action clues contained in other modal information to improve the performance. The comparison results of different methods in the table show that PromptLearner-CLIP achieves competitive results with state-of-the-art methods.

5 Conclusion

This paper proposes a multi-modal semantic-guided action recognition network PromptLearner-CLIP that utilizes textual information to enhance the representation ability of features. Experiments on Kinetics400, UCF101, and HMDB51 demonstrate that PromptLearner can automatically learn the text content of prompt and provide semantic clues for action recognition. Besides, by multi-modal information interaction module, features of different modalities pass information and disclose the difference of multi-modal features effectively. And

Table 7. Comparison with state-of-the-art methods on HMDB51 and UCF101 (%)

Model	HMDB51	UCF101
MSM-ResNets [54]	66.7	93.5
SVT [30]	67.2	93.7
two-stream TSN [39]	68.5	94.0
Temporal Squeeze Network[13]	71.5	95.2
TVNet+IDT [9]	72.6	95.4
TokenShift [48]	-	96.8
TEA-ResNet50 [23]	73.3	96.9
VidTr [49]	74.4	96.7
Dense Dilated Network [45]	74.5	96.9
Dynamic Network [51]	75.5	96.8
ActionCLIP[40]	76.2	97.1
BQN[14]	77.6	97.6
S3D-G [44]	78.2	96.8
MARS+RGB [6]	79.5	97.6
SIFP+SlowFast [20]	80.1	96.9
two-stream I3D [4]	80.2	97.9
PoTion+I3D [5]	80.9	98.2
STRM [32]	81.3	98.1
STA-MARS [21]	**81.4**	98.4
PromptLearner-CLIP+I3D(Flow)	81.3	**98.5**

the supervised contrastive loss is used to further reduce the feature distance between samples of the same class but different modalities. PromptLearner-CLIP achieves highly competitive accuracies on these three action recognition datasets with state-of-the-art methods. In future work, we will study the methods that incorporate the visual content of a video into prompt to better learn semantic information.

Acknowledgments This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFE0110500 and in part by the National Natural Science Foundation of China under Grant 62006015 and Grant 62072028..

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR. pp. 6077–6086. IEEE, Salt Lake City, UT, USA (2018)
2. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: a video vision transformer. In: ICCV. pp. 6836–6846. IEEE, Montreal, Canada (2021)
3. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML. pp. 813–824. ACM, Virtual (2021)

4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 4724–4733. IEEE, Honolulu, HI, USA (2017)
5. Choutas, V., Weinzaepfel, P., Revaud, J., Schmid, C.: Potion: pose motion representation for action recognition. In: CVPR. pp. 7024–7033. IEEE, Salt Lake City (2018)
6. Crasto, N., Weinzaepfel, P., Alahari, K., Schmid, C.: Mars: motion-augmented rgb stream for action recognition. In: CVPR. pp. 7874–7883. IEEE, Long Beach, CA, USA (2019)
7. 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 16x16 words: transformers for image recognition at scale. In: ICLR. pp. 1–21. Virtual (2021)
8. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: ICCV. pp. 6824–6835. IEEE, Montreal, Canada (2021)
9. Fan, L., Huang, W., Gan, C., Ermon, S., Gong, B., Huang, J.: End-to-end learning of motion representation for video understanding. In: CVPR. pp. 6016–6025. IEEE, Salt Lake City, UT, USA (2018)
10. Feichtenhofer, C.: X3d: expanding architectures for efficient video recognition. In: CVPR. pp. 200–210. IEEE, Seattle, WA, USA (2020)
11. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: CVPR. pp. 6202–6211. IEEE, Seoul, Korea (2019)
12. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9726–9735. IEEE, Seattle, WA, USA (2020)
13. Huang, G., Bors, A.G.: Learning spatio-temporal representations with temporal squeeze pooling. In: ICASSP. pp. 2103–2107. IEEE, Barcelona, Spain (2020)
14. Huang, G., Bors, A.G.: Busy-quiet video disentangling for video classification. In: WACV. pp. 1341–1350. IEEE, Waikoloa, HI, USA (2022)
15. Kar, A., Rai, N., Sikka, K., Sharma, G.: Adascan: adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. In: CVPR. pp. 5699–5708. IEEE, Honolulu, HI, USA (2017)
16. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset (2017)
17. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: NeurIPS. pp. 18661–18673. MIT Press, Virtual (2021)
18. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV. pp. 2556–2563. IEEE, Barcelona, Spain (2011)
19. 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: CVPR. pp. 7331–7341. IEEE, Virtual (2021)
20. Li, J., Wei, P., Zhang, Y., Zheng, N.: A slow-i-fast-p architecture for compressed video action recognition. In: ACM MM. pp. 2039–2047. ACM, Seattle, WA, USA (2020)
21. Li, J., Liu, X., Zhang, W., Zhang, M., Song, J., Sebe, N.: Spatiotemporal attention networks for action recognition and detection. IEEE Transactions on Multimedia **22**(11), 2990–3001 (2020)

22. Li, X., Wang, Y., Zhou, Z., Qiao, Y.: Smallbignet: integrating core and contextual views for video classification. In: CVPR. pp. 1092–1101. IEEE, Seattle, WA, USA (2020)
23. Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B., Wang, L.: Tea: temporal excitation and aggregation for action recognition. In: CVPR. pp. 906–915. IEEE, Seattle, WA, USA (2020)
24. Lin, J., Gan, C., Han, S.: Tsm: temporal shift module for efficient video understanding. In: ICCV. pp. 7082–7092. IEEE, Seoul, Korea (2019)
25. Liu, Z., Luo, D., Wang, Y., Wang, L., Tai, Y., Wang, C., Li, J., Huang, F., Lu, T.: Teinet: towards an efficient architecture for video recognition. In: AAAI. pp. 11669–11676. AAAI, New York, USA (2020)
26. Liu, Z., Wang, L., Wu, W., Qian, C., Lu, T.: Tam: temporal adaptive module for video recognition. In: ICCV. pp. 13708–13718. IEEE, Montreal, Canada (2021)
27. Neimark, D., Bar, O., Zohar, M., Asselmann, D.: Video transformer network. In: ICCV. pp. 3163–3172. IEEE, Montreal, Canada (2021)
28. Pan, T., Song, Y., Yang, T., Jiang, W., Liu, W.: Videomoco: contrastive video representation learning with temporally adversarial examples. In: CVPR. pp. 11200–11209. IEEE, Virtual (2021)
29. 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: ICML. pp. 8748–8763. ACM, Virtual (2021)
30. Ranasinghe, K., Naseer, M., Khan, S., Khan, F.S., Ryoo, M.: Self-supervised video transformer. In: CVPR. pp. 2874 – 2884. IEEE, New Orleans, Louisiana, USA (2022)
31. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: a dataset of 101 human actions classes from videos in the wild (2012)
32. Thatipelli, A., Narayan, S., Khan, S., Anwer, R.M., Khan, F.S., Ghanem, B.: Spatio-temporal relation modeling for few-shot action recognition. In: CVPR. pp. 19958 – 19967. IEEE, New Orleans, Louisiana, USA (2022)
33. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. pp. 4489–4497. IEEE, Santiago, Chile (2015)
34. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR. pp. 6450–6459. IEEE, Salt Lake City, UT, USA (2018)
35. Truong, T.D., Bui, Q.H., Duong, C.N., Seo, H.S., Phung, S.L., Li, X., Luu, K.: Direcformer: A directed attention in transformer approach to robust action recognition. In: CVPR. pp. 20030 – 20040. IEEE, New Orleans, Louisiana, USA (2022)
36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008. MIT Press, Long Beach, CA, USA (2017)
37. Wang, L., Li, W., Li, W., Van Gool, L.: Appearance-and-relation networks for video classification. In: CVPR. pp. 1430–1439. IEEE, Salt Lake City, UT, USA (2018)
38. Wang, L., Tong, Z., Ji, B., Wu, G.: Tdn: temporal difference networks for efficient action recognition. In: CVPR. pp. 1895–1904. IEEE, Virtual (2021)
39. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks for action recognition in videos. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(11), 2740–2755 (2019)
40. Wang, M., Xing, J., Liu, Y.: Actionclip: a new paradigm for video action recognition (2021)

41. Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Jiang, Y.G., Zhou, L., Yuan, L.: Bevt: Bert pretraining of video transformers. In: CVPR. pp. 14733 – 14743. IEEE, New Orleans, Louisiana, USA (2022)
42. Wang, X., Gupta, A.: Videos as space-time region graphs. In: ECCV. pp. 399–417. Springer, Munich, Germany (2018)
43. Wu, Z., Jiang, Y.G., Wang, X., Ye, H., Xue, X.: Multi-stream multi-class fusion of deep networks for video classification. In: ACM MM. pp. 791–800. ACM, Amsterdam, Netherlands (2016)
44. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification. In: ECCV. pp. 318–335. Springer, Munich, Germany (2018)
45. Xu, B., Ye, H., Zheng, Y., Wang, H., Luwang, T., Jiang, Y.: Dense dilated network for video action recognition. IEEE Transactions on Image Processing **28**(10), 4941–4953 (2019)
46. Yang, C., Xu, Y., Shi, J., Dai, B., Zhou, B.: Temporal pyramid network for action recognition. In: CVPR. pp. 588–597. IEEE, Seattle, WA, USA (2020)
47. Yang, S., Li, G., Yu, Y.: Dynamic graph attention for referring expression comprehension. In: ICCV. pp. 4643–4652. IEEE, Seoul, Korea (2019)
48. Zhang, H., Hao, Y., Ngo, C.W.: Token shift transformer for video classification. In: ACM MM. pp. 917–925. ACM, Chengdu, China (2021)
49. Zhang, Y., Li, X., Liu, C., Shuai, B., Zhu, Y., Brattoli, B., Chen, H., Marsic, I., Tighe, J.: Vidtr: video transformer without convolutions. In: ICCV. pp. 13577–13587. IEEE, Montreal, Canada (2021)
50. Zhao, Y., Wang, G., Luo, C., Zeng, W., Zha, Z.J.: Self-supervised visual representations learning by contrastive mask prediction. In: ICCV. pp. 10160–10169. IEEE, Virtual (2021)
51. Zheng, Y., Liu, Z., Lu, T., Wang, L.: Dynamic sampling networks for efficient action recognition in videos. IEEE Transactions on Image Processing **29**, 7970–7983 (2020)
52. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**, 2337–2348 (2022)
53. Zhu, L., Yang, Y.: Actbert: learning global-local video-text representations. In: CVPR. pp. 8746–8755. IEEE, Seattle, WA, USA (2020)
54. Zong, M., Wang, R., Chen, X., Chen, Z., Gong, Y.: Motion saliency based multi-stream multiplier resnets for action recognition. Image and Vision Computing **107**, 104108 (2021)