

Commonsense Knowledge Prompting for Few-shot Action Recognition in Videos

Yuheng Shi, Xinxiao Wu, *Member, IEEE*, Hanxi Lin, Jiebo Luo, *Fellow, IEEE*

Abstract—Few-shot action recognition in videos is challenging as the lack of supervision makes it extremely difficult to generalize well to unseen actions. To address this challenge, we propose a simple yet effective method, called knowledge prompting, which leverages commonsense knowledge of actions from external resources to prompt-tune a powerful pre-trained vision-language model for few-shot classification. To that end, we first collect a large-scale corpus of language descriptions of actions, defined as text proposals, to build an action knowledge base. The collection of text proposals is done by filling in a handcraft sentence template with an external action-related corpus or by extracting action-related phrases from captions of Web instruction videos. Next, we feed these text proposals to a pre-trained vision-language model along with video frames to generate matching scores of the proposals for each frame, and the scores can be treated as action semantics with strong generalization. Finally, we design a lightweight temporal modeling network to capture the temporal evolution of action semantics for classification. Extensive experiments on six benchmark datasets demonstrate that our method generally achieves state-of-the-art performance while reducing the training computational cost to 0.1% of the existing methods. Code is available at <https://github.com/OldStone0124/Knowledge-Prompting-for-FSAR>.

Index Terms—Few-shot action recognition; knowledge prompting; pre-trained vision-language model; action semantics

I. INTRODUCTION

Few-shot action recognition in videos aims to classify new action classes by using very few training samples. To solve this task, the majority of existing works [1]–[6] formulate the few-shot recognition problem in a meta-learning paradigm, where meta-metrics of similarity between actions are first trained in the training phase and then applied to the nearest neighbor voting to make predictions in the test phase. Although these methods have achieved promising performance on many datasets like Kinetics [7], they still suffer from the very scarcely labeled training data that limits their ability to generalize to seldom seen or even unseen action classes.

In this paper, we first present an insight that efficiently adapts a well-pre-trained vision-language model to solve the

few-shot action recognition task with minimal training. The motivation behind this insight is the superior generalization ability of a pre-trained vision-language model to novel tasks after it has seen massive image-text or video-text pairs during pre-training. Therefore, we propose a simple yet effective method, called knowledge prompting, which explores commonsense knowledge of actions from external resources to prompt-tune the pre-trained vision-language model effectively for few-shot recognition. In this work, we employ CLIP [8] as the pre-trained vision-language model.

To be more specific, we first build an action knowledge base by collecting large-scale textual descriptions of actions from external resources. These textual descriptions, namely text proposals, explicitly describe fine-grained movements of body parts (i.e., atomic actions) such as “human’s hand point to the tree” and “do a cartwheel”. To ensure that the knowledge base can cover as many action descriptions as possible, we propose two strategies to generate abundant and various text proposals. The first strategy uses a pre-defined sentence template to generate text proposals, where a sentence template of “subject-verb-object” is first created, and then the template is filled in with various action-related words from the external corpus. The corpus consists of the body motion concepts from the PaStaNet dataset [9] and the object categories from the Visual Genome dataset [10]. The text proposals generated in this way mainly describe basic actions and are used as a body of the knowledge base. The other strategy is designing a text proposal network that extracts action-related phrases from the captions of Web instruction videos to generate descriptions of daily actions, thus enriching the text proposals in the knowledge base.

Next, we take the text proposals and the video frames as inputs to the text encoder and the image encoder of CLIP, respectively, to learn action semantics for classification. For each frame, the output matching scores measure how similar the text proposals are to the visual content, and can be treated as potentially valuable representations of action semantics with strong generalization. Finally, we design a temporal modeling network to model the temporal context relationships between the proposal matching scores of different video frames, thereby capturing the evolution of action semantics over time for action classification. It should be emphasized that we keep the parameters of CLIP fixed in the training phase, and only train the lightweight temporal modeling network with very low computational cost. Extensive experiments on six benchmark datasets show that our method considerably boosts

Yuheng Shi and Hanxi Lin are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shiyuheng@bit.edu.cn).

Xinxiao Wu is with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China, and also with the Guangdong Provincial Laboratory of Machine Perception and Intelligent Computing, Shenzhen MSU-BIT University, Shenzhen 518172, China (e-mail: wuxinxiao@bit.edu.cn). Xinxiao Wu is the corresponding author.

Jiebo Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: jl原因@cs.rochester.edu).

the performance of few-shot action recognition on various datasets, while greatly reducing the training cost to less than 0.1 % of the existing methods.

The main contributions of our work are three-fold:

- We propose a knowledge prompting method that steers the pre-trained vision-language model (CLIP) to the few-shot action recognition task by leveraging commonsense knowledge from external resources. Our method is simple yet effective and has a strong generalization ability without expensive end-to-end training of a large-scale backbone.
- We propose two strategies to generate abundant and various textual descriptions of actions to build an action knowledge base, in order to effectively prompt CLIP for learning powerful representations of action semantics.
- We design a lightweight temporal modeling network to model the temporal evolution of action semantics, which further boosts recognition accuracy.

The remainder of this paper is organized as follows. In Section II, we summarize previous works related to our method. Section III describes the proposed knowledge prompting method for few-shot action recognition. Section IV discusses experimental results on various benchmark datasets, and the conclusion is given in Section V.

II. RELATED WORK

A. Pre-Trained Vision-Language Model for Recognition

Pre-trained vision-language models [8], [11] have achieved great success in visual recognition due to the addition of natural language to the supervised learning process. The core problem of applying these models to downstream tasks is prompt learning [12], which is a technique that seeks to exploit the learned knowledge encoded in a pre-trained model without tuning the model itself. Zhou *et al.* [13] propose to add learnable contexts to the text input of CLIP to learn task-relevant prompts for object recognition. Cho *et al.* [14] formulate several vision-and-language tasks in a unified generative architecture by fine-tuning the multi-modal pre-trained model using task-specific handcraft prompts. Tsimpoukelli *et al.* [15] train the vision model to learn to cooperate with the encoded common sense knowledge of the frozen language model to generate open-ended outputs and achieve few-shot learning.

In the field of action recognition, Wang *et al.* [16] use hand-crafted labels as the text input of CLIP [8] and fine-tune the whole pre-trained model. Wu *et al.* [17] present a two-stream framework that transfers bidirectional cross-modal knowledge from CLIP to enhance video recognition, an attribute branch leverages the video-to-text knowledge to generate attributes for auxiliary recognition, and a video branch uses the text-to-video expertise to generate temporal saliency to yield compact video representation. Different from the aforementioned methods, our knowledge prompting method takes full advantage of commonsense knowledge of actions and generates large-scale prompts to efficiently adapt the pre-trained CLIP model to few-shot action recognition, which no longer requires fine-tuning any parameter.

B. Few-shot Action Recognition

Many existing methods of few-shot action recognition concentrate on learning the transferable similarity metrics between actions for the nearest neighbor voting, due to the lack of training data. Some methods [1], [3]–[5] learn fine-grained video representations and use dot product or euclidean distance in the representation space as the similarity metric. Zhu *et al.* [3] propose a compound memory network to memorize key-frame features that are vital for adapting to new tasks. Perrett *et al.* [5] introduce a Transformer-like architecture to learn an adaptive representation (*i.e.* query-specific class prototype) via early fusion between the query video and support videos. Li *et al.* [18] summarize the drawbacks of metric learning as action duration and evolution misalignment, and address them sequentially through a two-stage network with temporal transformation, temporal rearrangement, and spatially offset prediction. Wu *et al.* utilize region representations with discriminative capability enhanced or adversarially train the learned latent features through a cross-view verification loss to explore effective representations for few-shot [19] or fully supervised [20] video-based person re-identification. There is also work on explicitly modeling the intrinsic property of video. Cao *et al.* [2] propose an ordered temporal alignment module to explicitly align video sequences using a variant of the dynamic time warping algorithm. Specifically, they design a deep distance measurement of the query video with respect to novel class proxies over its alignment path.

More recently, Zhu *et al.* [21] focus on exploiting the powerful pre-trained vision backbones rather than the meta-learning paradigm for few-shot action recognition. They present a classifier-based baseline method and fine-tune the pre-trained model to learn effective representations. In contrast, our method neither performs meta-learning nor fine-tunes vision backbones. It prompts the pre-trained vision-language model by leveraging external commonsense knowledge of actions to learn powerful action representations with the supervision of language.

III. OUR METHOD

A. Overview

We propose a knowledge prompting method for few-shot action recognition in videos. It prompts the pre-trained CLIP by using commonsense knowledge from external resources, thereby generalizing well to rare or even unseen actions. The commonsense knowledge is represented by textual descriptions of atomic actions, namely text proposals in this paper, and an action knowledge base is built by collecting text proposals from an external action-related corpus and video captions. The core issue of our method lies in how to collect rich and various text proposals for generating semantic representations of actions. To address this issue, we propose two strategies for collecting text proposals: handcraft generation via a sentence template and automatic generation via a text proposal network.

Given an input video, we first take the text proposals as the text input of CLIP, and take video frames as the image input of CLIP. Then, for each video frame, CLIP outputs the similarity

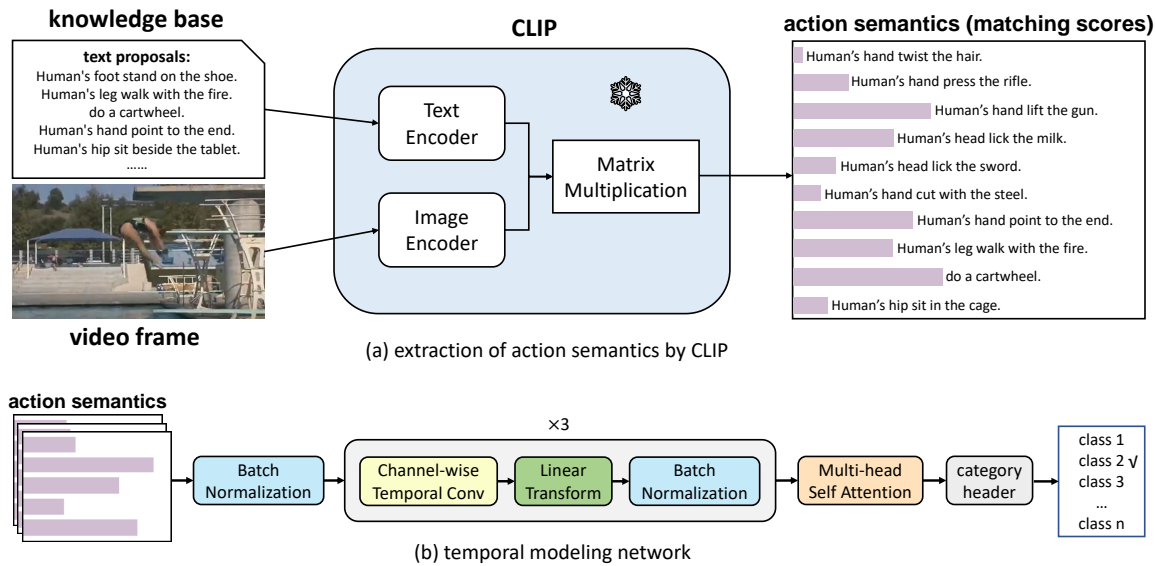


Fig. 1. Overview of the proposed method: a) Exaction of action semantics, where CLIP is used to extract action semantics by calculating the similarities between text proposals and video frames, and b) Temporal modeling network, where a sequence of action semantics is taken as the input and an action category label is predicted as the output.

matching scores of the text proposals that comprehensively describe the action semantics. Finally, we feed the matching scores of all the video frames into a newly designed temporal modeling network for action classification, by capturing the temporal evolution of action semantics. Fig. 1(a) shows the extraction of action semantics by CLIP, and Fig. 1(b) shows the temporal modeling network for classification. We only optimize the lightweight temporal modeling network and keep CLIP frozen, which achieves high computational efficiency in training.

B. Review of CLIP

CLIP (Contrastive Language-Image Pre-Training) [8] is a vision-language model pre-trained on millions of image-text pairs from Web by using a contrastive learning loss. CLIP consists of an image encoder and a text encoder, and predicts the matching score between the input image and text. It achieves incredible results of recognizing extensive visual concepts without manual labels and builds a reliable interaction between vision and text.

In recent years, CLIP has been successfully applied to a wide variety of downstream vision tasks, including image classification [13], [22], object detection [23], [24] and object navigation [25]. Consistent performance on these tasks validates the excellent generalization ability of CLIP, therefore we employ CLIP in this work for few-shot action recognition.

C. Generation of Text Proposals

Text proposals are actually abundant textual descriptions of atomic actions, and our action knowledge base is built by collecting text proposals from an external action-related corpus and video captions. The text proposals are generated by two strategies: (1) filling in a handcraft sentence template using

action-related words; (2) automatically generating from web video captions using a text proposal network.

1) *Handcraft Generation via Sentence Template*: The handcraft generation of text proposals is implemented by first creating a sentence template of “subject-verb-object” and then filling in the template using the action-related words from the external corpus. Although currently there is no corpus for directly describing human actions, there are still action-related datasets like PaStaNet [9] and Visual Genome [10]. So we use the body motion concepts from the PaStaNet dataset and the object categories from the Visual Genome dataset as the action-related corpus.

PaStaNet has a total of 93 states of 10 body parts, such as “hand, put on” and “head, kiss”, which provides subjects and verbs in the sentence template. Visual Genome has dense annotations of objects and scenes in images, and a total of 5,996 noun words or phrases in the annotations are selected as objects in the sentence template. In particular, all transitive verbs or phrases from PaStaNet are paired with nouns or noun phrases from Visual Genome, to fill in the sentence of “Human’s [body part] [state] the [object]”. For example, the body part state “foot, run to” and the noun “bed” are used to generate the text proposal “Human’s foot run to the bed”.

In this way, we generate 380,000 initial text proposals. However, they can not be directly fed into CLIP, since some linguistically unreasonable proposals will hurt the performance and the high dimension of matching score vector will make the computation very expensive. So we use a pre-trained mask-based language model, BERT [26], to filter the text proposals. To be more specific, we mask the object part (nouns) in the text proposals and use BERT to calculate the probabilities of the masked nouns according to the subject and the verb. If the probability is lower than a threshold λ (the value of λ will be analyzed in the experiments), the corresponding proposal will

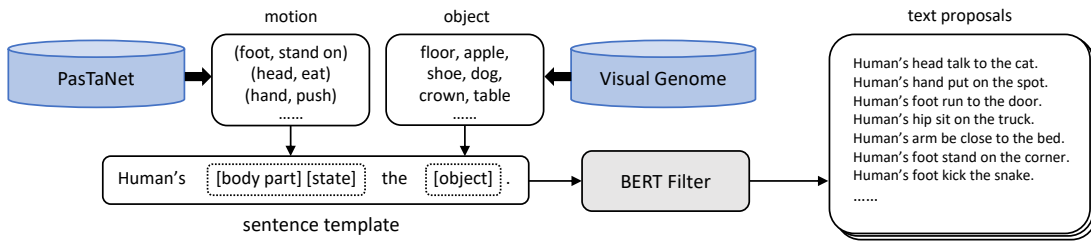


Fig. 2. Overview of the handcraft generation of text proposals via a sentence template. The action-related words from the PaStaNet and Visual Genome datasets are used to fill in the sentence template to generate initial text proposals, and then the BERT model is used to filter out linguistically unreasonable text proposals.

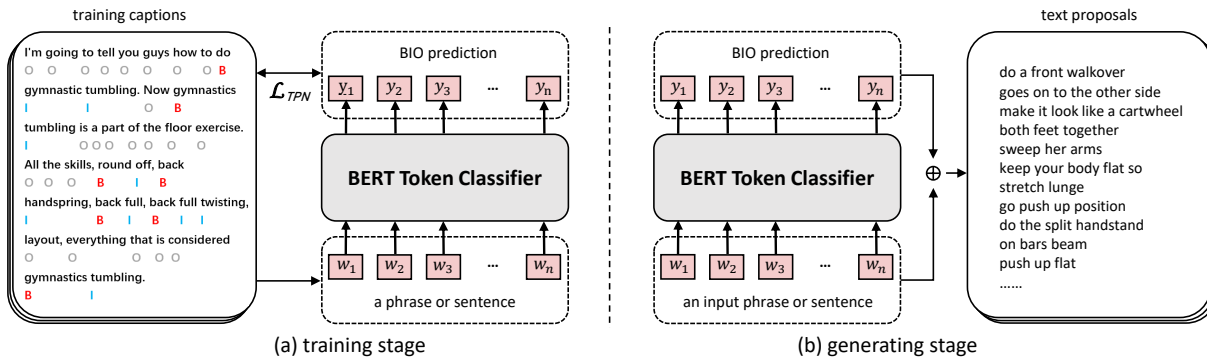


Fig. 3. Overview of the automatic generation of text proposals via text proposal network (TPN). TPN is actually a BERT token classifier, consisting of a BERT model and a fully connected layer. (a) Training stage: TPN is trained using the BIO-labeled captions. (b) Generation stage: text proposals are generated from input video captions via TPN.

be discarded. For example, for the masked proposal “Human’s foot stands on the [MASK]”, we tend to discard the nouns “code” and “license” with lower probabilities and adopt the nouns “bed” and “wood” with higher probabilities. Finally, we collect more than 50,000 text proposals as the main body of the knowledge base. Fig. 2 illustrates the process of the handcraft generation of text proposals via sentence template.

2) *Automatic Generation via Text Proposal Network*: To generate more diverse text proposals to further improve the scalability of the knowledge base, we propose a text proposal network (TPN) that automatically extracts text proposals of daily actions from the action-related captions of Web instruction videos. It takes video captions as input and outputs action description phrases as the text proposals.

To collect the captions of instruction videos from the Web, we use query keywords like “how to”, “tutorial” and “teach” to search action-related instruction videos such as diving and gymnastics tutorial videos from Youtube, and crawl the corresponding captions that have abundant action descriptions. We sample 10 captions with about 50,000 words as the training data and annotate the words using the BIO format annotation method [27]. Specifically, for each action description (i.e., phrase or sentence), “B” is used to label the first word, and “I” is used to label the remaining words. For other descriptions that do not describe actions, “O” is used to label the words.

TPN consists of a BERT model to extract the token feature of the input sentence, and a fully connected layer to judge whether or not a token belongs to the output text proposal. Specifically, it classifies the input words into three classes:

the first words of action descriptions (“B”), the remaining words of action descriptions (“I”), and the words of non-action descriptions (“O”). The training captions with their corresponding BIO annotations are used to train TPN, and a cross-entropy loss is used for model optimization. Fig. 3(a) shows the training of TPN. Given a training caption of n words with their corresponding ground-truth labels $\{a_1, a_2, \dots, a_n\}$, the cross entropy loss for word classification is defined as

$$\mathcal{L}_{TPN}(\theta) = -\frac{1}{n} \sum_{i=1}^n \sum_{c \in C} y_{ic} \log \hat{y}_{ic} \quad (1)$$

where C denotes the label set $\{“B”, “I”, “O”\}$ and θ denotes the parameters of BERT and the fully connected layer. y_{ic} represents the ground-truth label of the word w_i , formulated by

$$y_{ic} = \begin{cases} 1, & \text{if } a_i = c \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

We fine-tune the entire model during training. For the training details, we apply a method named BertForTokenClassification, which replaces the pooling layer of original BERT with a classification layer, since each word has to be predicted for ‘BIO’ classification.

In the generating stage, we use the trained TPN to classify the words of an input caption into “B”, “I” or “O”, and take the words predicted as “B” or “I” as the output text proposals. Fig. 3(b) shows the generation of text proposals using TPN. By applying the trained TPN to the instruction video captions,

we generate about 4,000 text proposals which further enriches the knowledge base.

D. Temporal Modeling of Action Semantics

The generated text proposals are taken as the input of the text encoder in CLIP, and the video frames are fed into the image encoder in CLIP. The output similarity matching scores between the text proposals and each frame actually represent the action semantics of the frame, owing to the great potential of CLIP in bridging the two modalities of vision and language. To capture the temporal relationships between action semantics of different video frames for classification, we propose a temporal modeling network that integrates temporal convolution and multi-head self-attention.

1) *Extraction of Action Semantics*: Given an input video with n frames $\{f_1, f_2, \dots, f_n\}$, and a set of m text proposals $\{p_1, p_2, \dots, p_m\}$, CLIP calculates the matching similarities between the frames and the proposals, denoted as $\mathbf{S} \in \mathbf{R}^{n \times m}$, where \mathbf{S}_{ij} represents the matching score between the i -th video frame f_i and the j -th text proposal p_j . The higher \mathbf{S}_{ij} is, the more relevant p_j is to f_i . The similarity matching scores represent how the corresponding textual descriptions of actions relate to the frames, and thus can be treated as the action semantics of the frames. Let $\mathbf{v}_i = [\mathbf{S}_{i1}, \mathbf{S}_{i2}, \dots, \mathbf{S}_{im}]$ denote the i -th row of \mathbf{S} , and it represents the action semantics of the i -th frame. Since the collected text proposals cover rich and various descriptions of atomic actions, the action semantics are more like intermediate-level representations of actions with strong generalization.

It is worth mentioning that the extraction of action semantics does not require training any parameter of CLIP, and we only need to perform the extraction process once for each sample and store the action semantics offline during training. This differs from other previous methods [2], [5], [16] that require complete forward and back propagation using the backbone network for each sample in each iteration. Therefore, our method maintains an extremely low computational cost while achieving state-of-the-art few-shot action recognition performance.

2) *Temporal Modeling Network*: To capture the temporal contextual relationships between the action semantics to further improve the recognition performance, we design a lightweight temporal modeling network (TMN), in which the action semantics are scaled, combined, time-series modeled, and finally mapped to the action category space.

As illustrated in Fig. 1(b), TMN mainly consists of a batch normalization layer, multiple channel-wise temporal convolution layers, and a multi-head self-attention module. Given an input sequence of action semantics $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L\}$, where \mathbf{v}_i is the action semantics of the i -th video frame, the batch normalization layer is first employed to eliminate the distribution bias of CLIP for fitting the prior distribution to its training data. Then the multiple channel-wise temporal convolutions with linear transformation and batch normalization are applied for the temporal modeling of action semantics. In the multiple channel-wise temporal convolutions, we perform 1D-CNNs on the frame sequence, including a 1×1 kernel to

reduce the semantic feature channels for each frame and a 1×3 kernel to extract local temporal feature between adjacent frames. Finally, the multi-head self-attention module is used for global temporal modeling of the features of all frames, and a linear category header is used for classification. We use a cross-entropy loss to train TMN.

IV. EXPERIMENTS

A. Datasets

We conduct experiments on six action datasets for evaluation, including Kinetics [7], Something Something V2 (SS-V2) [33], HMDB51 [34], UCF101 [35], Diving48-V2 [36], and FineGym [37].

Kinetics [7] is a large-scale high-quality dataset of YouTube videos with various human actions, including human-object interactions such as playing a musical instrument, and human-to-human interactions such as shaking hands. We adopt the few-shot version [3] that contains 100 categories selected from Kinetics, where the categories are split into 64/12/24 for train/validation/test sets, respectively, and there are 100 videos for each category.

SS-V2 [33] is a large collection of labeled videos that record fine-grained actions between human hands and objects. It contains 220,847 densely-labeled videos covering 174 categories. We take the existing splits proposed in [2], where 64/12/14 categories with 77,500/1,925/2,854 videos are in train/validation/test sets, respectively.

HMDB51 [34] is a relatively small dataset with 51 categories, and most videos come from movies or public datasets. The actions mainly include facial movements, face-object interactions, body movements, body-object interactions and human-human interactions. We follow the same protocol introduced in [4] which takes 31/10/10 categories for train/validation/test sets, respectively, and each category has at least 100 videos.

UCF101 [35] is a popular action recognition dataset of realistic action videos collected from YouTube. It has 13,320 videos covering 101 categories. We also follow the same protocol introduced in [4], where 70/10/21 categories with 9,154/1,421/2,745 videos are for train/validation/test sets, respectively.

Diving48-V2 [36] is a fine-grained action dataset and consists of diving videos. It has 18,000 videos covering 48 categories and includes four attributes (takeoff, somersaults, twists and dive). We take 36/6/6 categories for train/validation/test sets, respectively.

FineGym [37] is a large-scale and hierarchically labeled fine-grained action dataset built on gymnasium videos. It provides temporal annotations at both action and sub-action levels with a three-level (event, set, element) semantic hierarchy. We use the element-level actions with 99 categories for experiments, and take 72/13/14 categories for train/validation/test sets, respectively.

B. Implement Details

During training, CLIP is frozen and only the temporal modeling network is trained. During test, the parameters of

TABLE I

COMPARISON RESULTS (%) BETWEEN DIFFERENT METHODS OF 5-WAY 5-SHOT ON THE KINETICS, SS-V2, HMDB51, UCF101, DIVING48-V2 AND FINEGYM DATASETS. THE "BACKBONE" COLUMN REPRESENTS WHETHER THE METHOD HAS A BACKBONE TO BE TRAINED.

Method	Backbone(trained)	Kinetics	SS-V2	HMDB51	UCF101	Diving48-V2	FineGym
TARN [1]	C3D(✓)	80.7	-	-	-	-	-
ARN [4]	C3D(✓)	82.4	-	60.6	83.1	-	-
TARN [1]	ResNet-50(✓)	78.5	-	-	-	-	-
CMN [3]	ResNet-50(✓)	78.9	-	-	-	-	-
CMN-J [28]	ResNet-50(✓)	78.9	-	-	-	-	-
OTAM [2]	ResNet-50(✓)	85.8	52.3	72.1	-	56.2	67.3
TA2N [18]	ResNet-50(✓)	85.8	61.0	73.9	95.1	-	-
TRX [5]	ResNet-50(✓)	85.9	64.6	75.6	96.1	62.6	72.7
STRM [29]	ResNet-50(✓)	86.7	68.1	77.3	96.9	-	-
MTFAN [30]	ResNet-50(✓)	87.4	60.4	74.6	95.1	-	-
HyRSM [31]	ResNet-50(✓)	86.1	69.0	76.0	94.7	-	-
STRM [29]	ViT-B(✓)	91.2	70.2	81.3	98.1	-	-
CLIP [8]	ViT-B(X)	94.2	26.2	31.3	93.5	22.9	20.1
CLIP with TMN	ViT-B(X)	91.7	56.0	84.7	99.0	78.8	74.0
Ours	ResNet-50(X)	90.4	57.0	83.6	99.1	79.6	76.4
Ours	ViT-B(X)	94.3	62.4	87.4	99.4	82.6	76.8

TABLE II

COMPARISON RESULTS (%) BETWEEN DIFFERENT METHODS OF 5-WAY 1-SHOT ON THE KINETICS, SS-V2, HMDB51, UCF101, DIVING48-V2 AND FINEGYM DATASETS. THE "BACKBONE" COLUMN REPRESENTS WHETHER THE METHOD HAS A BACKBONE TO BE TRAINED.

Method	Backbone(trained)	Kinetics	SS-V2	HMDB51	UCF101	Diving48-V2	FineGym
TARN [1]	C3D(✓)	66.6	-	-	-	-	-
ARN [4]	C3D(✓)	63.7	-	45.5	66.3	-	-
TARN [1]	ResNet-50(✓)	64.8	-	-	-	-	-
CMN [3]	ResNet-50(✓)	60.5	-	-	-	-	-
OTAM [2]	ResNet-50(✓)	73.0	42.8	-	-	53.5	61.5
TA2N [18]	ResNet-50(✓)	72.8	47.6	59.7	81.9	-	-
TRX [5]	ResNet-50(✓)	63.6	42.0	-	-	41.3	62.3
MTFAN [30]	ResNet-50(✓)	74.6	45.7	59.0	84.8	-	-
HyRSM [31]	ResNet-50(✓)	73.7	54.3	60.3	83.9	-	-
Huang <i>et al.</i> [32]	ResNet-50(✓)	73.3	49.3	60.1	71.4	-	-
CLIP [8]	ViT-B(X)	94.2	26.2	31.3	93.5	22.9	20.1
CLIP with TMN	ViT-B(X)	82.0	41.8	73.4	94.6	67.0	66.2
Ours	ResNet-50(X)	78.0	41.9	67.7	92.8	65.0	67.8
Ours	ViT-B(X)	85.2	44.7	75.8	97.4	68.1	68.6

both CLIP and the temporal modeling network are kept fixed, except for the linear category header that is fine-tuned for the target classes. ViT-B/16 [38] is used as the image encoder and the text encoder of CLIP. The pre-processing of image and text feature extraction remains the same as the original CLIP. The temporal sparse sampling [39] is adopted to sample video frames as the input of CLIP, and the number of sampled frames is set to 16 for each video.

In terms of model training, SGD with momentum [40] is used as the optimizer. The learning rate is initially set to 0.001 and is attenuated by 10 times at 20, 30, and 40 training epochs, respectively. A random dropout layer with a dropout probability of 0.05 is applied after the first batch normalization layer of the temporal modeling network. The momentum coefficient is 0.9, the L2 regularization coefficient is 0.001, and the batch size is 32.

In terms of test settings, Adam [41] is used as the optimizer.

The initial learning rate is 0.01, and the exponential decay rate coefficients for moment estimates are set to 0.5 and 0.999. The training stops after 10 training epochs. The L2 regularization coefficient is 0 and the batch size is 16. The prediction result of a single sample is the average prediction result on 10 random samplings of the video in a temporal sequence, as the number of frames in each video sample is about 150 to 200 and we sample 16 frames from each video with the percentage of frames in a certain range (8% to 11%). The standard few-shot evaluation is employed on all datasets, taking 5-way 5-shot for example, we randomly select 5 categories from the test data (5-way), and randomly select 10 videos from each category, of which 5 videos serve as the support set (5-shot) and the other 5 videos serve as the query set. The average accuracy over 500 random test tasks is reported, following TARN [1].

TABLE III
RESULTS (%) OF ABLATION STUDIES ON THE KINETICS, SS-V2, HMDB51, UCF101, DIVING48-V2 AND FINEGYM DATASETS.

Method	Kinetics	SS-V2	HMDB51	UCF101	Diving48-V2	FineGym
w/o knowledge	91.7	56.0	84.7	99.0	78.8	74.0
w/o template	90.0	58.8	83.0	98.0	80.0	74.5
w/o TPN	94.1	62.2	86.8	99.6	81.5	76.1
w/o TMN	93.3	49.0	85.2	99.2	63.7	69.0
Ours	94.3	62.4	87.4	99.4	82.6	76.8

TABLE IV
RESULTS (%) OF USING THE TEXT PROPOSALS GENERATED BY HANDCRAFT SENTENCE TEMPLATE WITH DIFFERENT λ ON THE KINETICS, SS-V2, HMDB51, UCF101, DIVING48-V2 AND FINEGYM DATASETS.

Value of λ	Proposal Number	Kinetics	SS-V2	HMDB51	UCF101	Diving48-V2	FineGym
6×10^{-4}	14388	92.1	60.0	85.1	99.0	80.1	74.8
3×10^{-4}	25172	93.0	61.3	86.1	99.1	80.8	75.0
2×10^{-4}	33763	93.5	61.6	85.3	99.2	80.7	76.1
1×10^{-4}	53133	94.1	62.2	86.8	99.6	81.5	74.9

TABLE V
RESULTS (%) OF ABLATION STUDIES OF DIFFERENT INPUT FRAME NUMBERS FOR 5-WAY 5-SHOT ON THE KINETICS, HMDB51 AND DIVING48-V2 DATASETS.

Backbone	Frames	Kinetics	HMDB51	Diving48-V2
ViT/B	8	94.0	87.0	77.9
ViT/B	16	94.3	87.4	82.6
ViT/B	32	94.2	86.9	81.4
ResNet-50	8	89.9	83.4	73.9
ResNet-50	16	90.4	83.7	79.6
ResNet-50	32	90.2	83.6	69.9

TABLE VI
RESULTS (%) OF DIFFERENT DATASETS IN HANDCRAFT GENERATION OF TEXT PROPOSALS FOR 5-WAY 5-SHOT ON THE KINETICS, HMDB51 AND DIVING48-V2 DATASETS.

Text Proposal	Nouns	Kinetics	HMDB51	Diving48-V2
Visual Genome	5996	94.1	87.4	81.5
ImageNet-1k	1000	90.4	81.9	78.4

C. Experimental Results

1) *Comparison with State-of-the-Art Methods:* Table I and II shows the comparison results of the 5-way 5-shot and 5-way 1-shot evaluation with the state-of-the-art methods on the six action datasets. It can be observed that our method generally achieves the best results on most datasets. This benefits from the strong generalization of extracted action semantics by using the collection of abundant text proposals and the powerful vision-language matching ability of CLIP. Moreover, we compare our method with OTAM and TRX in terms of training computational cost and inference speed of model. The results are reported in Table VII, which shows that the computational cost of our method is extremely low, mainly due to the frozen CLIP parameters and the lightweight temporal modeling network. We also observe that the inference time of our method is slower than OTAM and TRX, as we use the support data to fine-tune the 5-way linear classifier during each test task. The experiments are carried out on one NVIDIA GeForce RTX 3090 GPU.

We also observe that the proposed method performs not

TABLE VII
COMPARISON RESULTS (%) OF COMPLEXITY STUDIES FOR 5-WAY 5-SHOT ON THE DIVING48-V2 DATASET.

Method	Backbone(trained)	FLOPs	Inference Time
OTAM	ResNet-50(✓)	994.3	112.0ms
TRX	ResNet-50(✓)	1026.7	105.9ms
Ours	ResNet-50(✓)	0.8	193.6ms

very well on the SS-V2 dataset, probably due to that most of the actions in SS-V2 are about fine-grained hand-object interactions, such as “pretending to put something underneath something” and “moving something across a surface until it falls down”, and it is extremely difficult to collect relevant descriptions of these actions from external resources as text proposals.

2) *Comparison with Baseline Method:* We compare our method with a baseline method, called CLIP, which directly uses only the visual features extracted by the image encoder of CLIP for action recognition without temporal modeling. We also introduce CLIP with the temporal modeling network (CLIP with TMN) for comparison, where TMN follows the CLIP image encoder for few-shot action recognition.

The results are shown in the bottom part of Table I. We can observe that our method achieves better results than CLIP on all the datasets, especially on SS-V2, Diving48-V2 and FineGym, which demonstrates the superiority of extracting action semantics via prompting CLIP using commonsense knowledge and modeling the temporal information of action semantics by TMN.

We also observe that CLIP with TMN performs worse than CLIP on Kinetics, probably due to that the data distribution of Kinetics is relatively close to that of CLIP’s pre-training data, i.e., they both are common images and video screenshots collected from Internet, which have the characteristics of weak temporal variation. Therefore, CLIP can take advantage of its strong generalization and achieve better results.

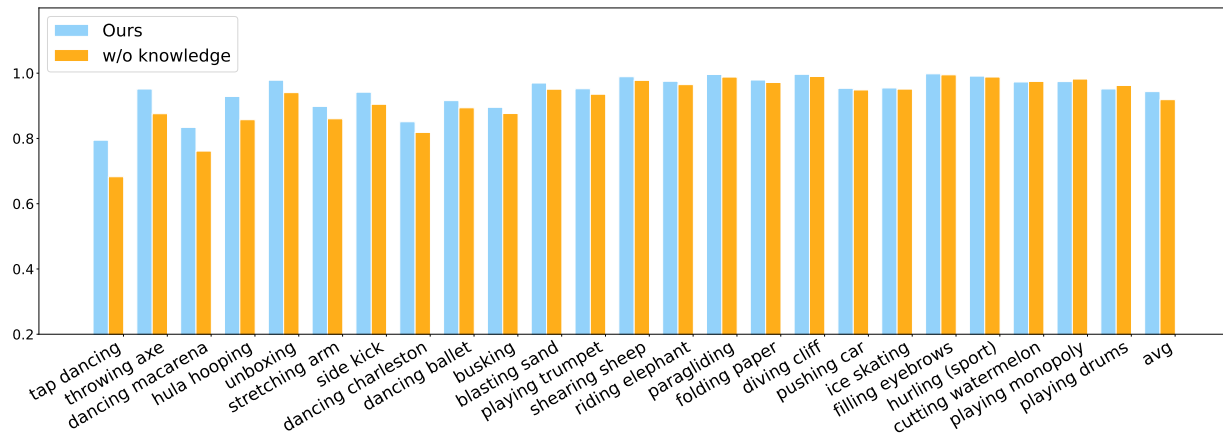


Fig. 4. Results comparison between “w/o knowledge” and our method on different categories in the Kinetics dataset. The horizontal axis indicates the action category label and the vertical axis indicates the standard 5-way 5-shot recognition accuracy.

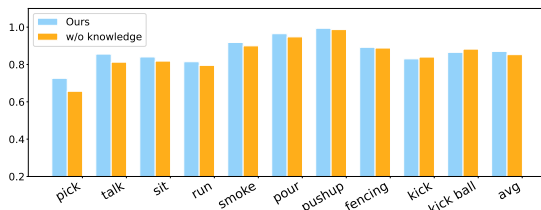


Fig. 5. Comparison results between “w/o knowledge” and our method on different categories in the HMDB51 dataset. The horizontal axis indicates the action category label and the vertical axis indicates the standard 5-way 5-shot recognition accuracy.

D. Ablation Studies

To study different individual components in-depth, we introduce several variants of our method for comparison as follows.

- **w/o knowledge:** The knowledge base is removed to evaluate the contribution of text proposals. In this case, only the visual features from the image encoder of CLIP are directly fed into the temporal modeling network for classification.
- **w/o template:** The handcrafted text proposals using the sentence template are removed to evaluate their effectiveness. In this case, only the text proposals generated by the text proposal network are used to extract action semantics for classification.
- **w/o TPN:** The automatically generated text proposals using the text proposal network are removed to evaluate their effectiveness. In this case, only the text proposals generated by the sentence template are used.
- **w/o TMN:** The temporal modeling network is replaced by a linear mapping layer along with a batch normalization layer to evaluate its importance to classification.

The results of ablation studies on the six datasets are shown in Table III. We have the following observations:

- The performance degrades on all the datasets when removing the text proposals, which validates the benefit of prompting CLIP using external knowledge to enhance the generalization ability in few-shot recognition.

- When removing the text proposals generated by the sentence template, our model shows performance degradation on all datasets. Such results clearly verify the strong generalization ability of the template-generated text proposals and their dominance in the knowledge base.
- When removing the text proposals automatically generated by TPN, the performance also drops on most datasets, which indicates the efficacy of extracting action descriptions from the captions of Web instruction videos on enriching the knowledge base of text proposals. It should be mentioned that the proposals generated by TPN will inevitably contain some noise words since some video captions are automatically generated by speech recognition, but experiments show that this does not affect their vital role in recognition.
- When removing TMN, our method achieves much worse results, clearly demonstrating that it is essential to capture the temporal relationships between action semantics for action classification.

Fig. 4 and 5 show the comparison results between “w/o knowledge” and our method of different action categories on the Kinetics and HMDB51 datasets, respectively. It is evident that our method improves the recognition accuracy of most categories on the two datasets, which suggests that the collection of a large amount of high versatility action description proposals enables the generation of more granular action semantics for video frames than traditional visual features, thereby enhancing the capacity to distinguish between different categories. We also observe that our method with knowledge does not perform very well on a few categories, such as “kick” and “kick ball” on the HMDB51 dataset, and this may be due to the fact that these categories have similar physical motions and so their action semantics are too similar to distinguish them.

To further analyze the effectiveness of the proposed temporal modeling network (TMN), we visualize the features of action semantics before (“w/o TMN”) and after TMN (“our method”) using t-SNE [42] on the Kinetics and HMDB51

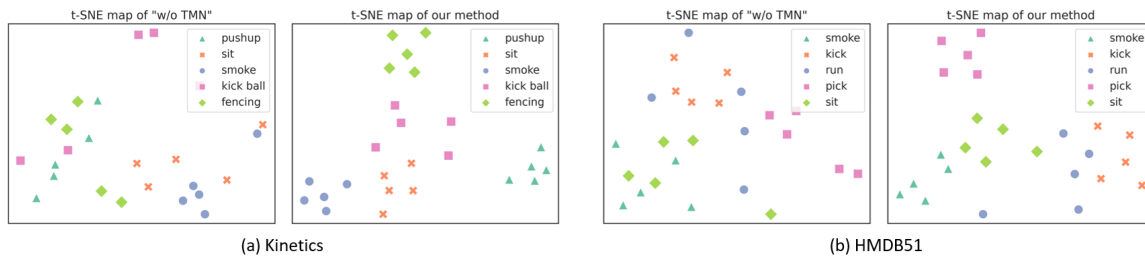


Fig. 6. Feature visualization comparisons between “w/o TMN” and our method on the Kinetics and HMDB51 datasets using t-SNE. Five samples from each category are shown and five categories are shown. Different colors and shapes represent different categories.

datasets in Fig. 6. In the 5-way 5-shot setting, five samples from each category are shown and five categories are shown. We can observe that the features learned by TMN are more discriminative for classification, which verifies its superiority in precisely capturing the temporal relationships between action semantics to improve the action recognition performance.

E. Analysis of Threshold λ

To analyze the effect of text proposal filtering in handcraft generation, we evaluate the effectiveness of different values of the probability threshold λ in filtering out text proposals. Table IV shows the results of only using the text proposals generated by handcraft sentence template with different λ , where the larger λ represents that more text proposals are filtered out. It can be observed that a smaller λ generally achieves fairly better performance, which suggests that the increasing text proposals are helpful to boost the accuracy owing to more supervision from language. For FineGym, the optimal value of λ is larger than that for other datasets. The possible reason is that the videos in the FineGym dataset are professional actions of formal gymnastics competitions where the scenes and interactive objects are relatively simple compared to other datasets, so there are fewer proposals related to frames. In this case, more proposals with low probability bring more distractors and hurt the performance.

F. Analysis of different datasets in handcraft generation of text proposals

To analyze the impact of different datasets in handcraft generation of text proposals, we use ImageNet-1k that contains a diverse range of 1000 object categories for comparison. Specifically, as there are no other available part-level body motion datasets, we use the body motion concepts from PaStaNet and compare the performance of using object categories from Visual Genome and ImageNet-1k. It can be observed in Table VI that the text proposals generated by Visual Genome achieve better performance than that generated by ImageNet-1k, which indicates that the rich variety of object categories in Visual Genome enhances the generalization capability of text proposals.

G. Exploration of the number of input frames

To explore the impact of the number of input video frames, we compare the performance of our method using different

input frame numbers (8, 16 and 32) on Kinetics, HMDB51 and Diving48-V2, and the results are reported in Table V. It can be observed that our method performs best with 16 input frames. Using 8 frames as input would lack motion information, while using 32 frames would lead to overfitting on the training set.


H. Evaluation of Text Proposals

Fig. 7 illustrates several examples of video frames with top 10 important text proposals before and after TMN, where the importance of text proposals before TMN is determined by the matching similarity scores after CLIP, and that after TMN is calculated by the gradient values of the first batch normalization layer in TMN.


Fig. 7(a) shows an example of the action “stretching arm” from the Kinetics dataset. It can be seen that the top important text proposals before TMN are already sufficient for recognizing the action “stretching arm” because they describe the motion of “stretch”. However, it is more interesting to observe that the top text proposals after TMN change to pay more attention to moving body parts besides the arm and improve the classification probability. It suggests that the motion information of multiple body parts is essential to action recognition, which is successfully captured by TMN.

Fig. 7(b) shows an example of the action “pick” from the HMDB51 dataset, which further interprets the benefit of TMN to making correct predictions. This action example is about a man picking up rubbish from the roadside to his bag. But before TMN, the shapes of man and bag are somewhat misleading, making CLIP recognize a motorcycle in the frames and resulting in high scores for the text proposals describing “run” or “motorcycle” to make the wrong prediction of “run”. Our method makes the correct classification of “pick”, and it can be seen that most important proposals after TMN are closer to dynamically describing the actions like “Human’s hand write on the belt.” and “Human’s hand throw the bag”. This is due to the reliability of TMN and the strong generalization ability of the collected abundant text proposals.


Fig. 7(c) shows an example of the action “transition flight from high bar to low bar” on the FineGym dataset. Most initially generated text proposals before TMN mainly describe the interaction between “human” and “beam” or “crossbar”. It is because the “beam” or “crossbar” is really the most obvious object in video frames, which is easily recognized by CLIP without temporal modeling. After TMN, it is interesting to observe that some proposals like “getting your feet together

Video frames	Before TMN	After TMN
	<p>Action category: stretching arm (0.921)</p> <p>Text proposals (top 10):</p> <ol style="list-style-type: none"> 1. a backbend lay flat push up 2. stretch lunge 3. go push up position 4. performing these simple stretches 5. use this tuck shape for our forward roll 6. push up flat 7. do a one-handed cartwheel 8. start a backbend kickover 9. does the split handstand 10. a cool bit easy gymnastics move 	<p>Action category: stretching arm (0.948)</p> <p>Text proposals (top 10):</p> <ol style="list-style-type: none"> 1. Human's hip sit beside the hilt. 2. Human's leg is close with the room. 3. Human's hand carry the flesh. 4. Human's arm hug the cross. 5. Human's hand point with the scope. 6. Human's shoulder carry the equipment. 7. reverse it 8. Human's head wear the headband. 9. Human's hip sit in the machine gun. 10. Human's head be close with the gesture.

(a) The action "stretching arm" on the Kinetics dataset.

Video frames	Before TMN	After TMN
	<p>Action category: run (0.403)</p> <p>Text proposals (top 10):</p> <ol style="list-style-type: none"> 1. Human's leg run to the motorcycle. 2. Human's leg walk with the motorcycle. 3. Human's leg run with the motorcycle. 4. Human's hand throw out the street. 5. Human's leg walk to the motorcycle. 6. Human's hand throw out the road. 7. Human's head blow the road. 8. Human's foot walk to the motorcycle. 9. Human's foot run to the town. 10. Human's foot run to the bus stop. 	<p>Action category: pick (0.706)</p> <p>Text proposals (top 10):</p> <ol style="list-style-type: none"> 1. Human's hand write on the belt. 2. Human's hand throw out the light bulb. 3. Human's head kiss the ground floor. 4. Human's leg walk with the owner. 5. Human's hand throw the bag. 6. Human's hand twist the cup. 7. Human's arm be close to the chip. 8. Human's head be close with the arrow. 9. Human's hand throw out the signal. 10. Human's leg run with the town.

(b) The action "pick" on the HMDB51 dataset.

Video frames	Before TMN	After TMN
	<p>Action category: transition flight from high bar to low bar (0.59)</p> <p>Text proposals (top 10):</p> <ol style="list-style-type: none"> 1. on bars beam 2. good pivot turn at the end of the beam 3. jump to the high bar swing 4. Human's hand raise the crossbar. 5. Human's hand catch with the beam. 6. Human's leg jump with the beam. 7. Human's leg jump with the crossbar. 8. Human's foot jump with the beam. 9. see a oh big release handspring out 10. in the gymnasts routine 	<p>Action category: transition flight from high bar to low bar (0.975)</p> <p>Text proposals (top 10):</p> <ol style="list-style-type: none"> 1. getting your feet together ham 2. have the chin up 3. with an atomic arm up Anton 4. with your hands here at 5. weight distribution coming off your arms 6. with hands in position 7. Human's hip sit beside the jamb. 8. Human's hand pour into the matchbox. 9. keeping really strong and 10. Human's hand throw out the bookshop.

(c) The action "transition flight from high bar to low bar" on the FineGym dataset.

Fig. 7. Several examples of actions with top 10 important text proposals before and after the temporal modeling network (TMN) on the Kinetics, HMDB51 and FineGym datasets. The number in the bracket after each action category label represents the probability of classifying the video into the corresponding category. The most discriminative text proposals for action recognition are marked in green. The incorrectly classified category label is marked in red.

ham” and “with an atomic arm up Anton” become more important, since they describe the discriminative fine-grained body movements and thus play a vital role in final recognition.

V. CONCLUSION

We have presented a knowledge prompting method that can efficiently adapt a pre-trained vision-language model (CLIP) by leveraging commonsense knowledge from external resources to achieve the few-shot action recognition. To that end, we have proposed two strategies that are able to generate abundant text proposals as the text input of CLIP. A lightweight network is also designed for temporal modeling of action semantics and succeeds in boosting performance. Our method is simple yet effective, with a strong generalization ability and

low computational cost. Extensive experiments on six action datasets demonstrate the effectiveness and superiority of our method on few-shot action recognition.

For some specific actions such as fine-grained hand movements in the SS-V2 dataset, the performance of our method is not satisfactory due to the limited relevant text proposals. So in future work, we are going to explore more external resources to further enrich the knowledge base, and meanwhile introduce uncertainty learning to improve the text proposal prompting.

VI. ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62072041.

REFERENCES

[1] M. Bishay, G. Zoumpourlis, and I. Patras, “Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition,” *arXiv preprint arXiv:1907.09021*, 2019.

[2] K. Cao, J. Ji, Z. Cao, C.-Y. Chang, and J. C. Niebles, “Few-Shot Video Classification via Temporal Alignment,” in *Computer Vision and Pattern Recognition*, 2020.

[3] L. Zhu and Y. Yang, “Compound Memory Networks for Few-Shot Video Classification,” in *European Conference on Computer Vision*, 2018.

[4] H. Zhang, L. Zhang, X. Qi, H. Li, P. H. S. Torr, and P. Koniusz, “Few-shot Action Recognition with Permutation-invariant Attention,” in *European Conference on Computer Vision*, 2020.

[5] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, and D. Damen, “Temporal-relational crosstransformers for few-shot action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 475–484.

[6] R. Ben-Ari, M. S. Nacson, O. Azulai, U. Barzelay, and D. Rotman, “Taen: Temporal aware embedding network for few-shot action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2786–2794.

[7] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[9] Y.-L. Li, L. Xu, X. Liu, X. Huang, Y. Xu, S. Wang, H.-S. Fang, Z. Ma, M. Chen, and C. Lu, “Pastanet: Toward human activity knowledge engine,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 382–391.

[10] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *International journal of computer vision*, vol. 123, no. 1, pp. 32–73, 2017.

[11] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, “Scaling up visual and vision-language representation learning with noisy text supervision,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 4904–4916.

[12] T. Schick and H. Schütze, “Exploiting cloze questions for few shot text classification and natural language inference,” *arXiv preprint arXiv:2001.07676*, 2020.

[13] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, “Learning to prompt for vision-language models,” *arXiv preprint arXiv:2109.01134*, 2021.

[14] J. Cho, J. Lei, H. Tan, and M. Bansal, “Unifying vision-and-language tasks via text generation,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 1931–1942.

[15] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, “Multimodal few-shot learning with frozen language models,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 200–212, 2021.

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

[17] W. Wu, X. Wang, H. Luo, J. Wang, Y. Yang, and W. Ouyang, “Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models,” *arXiv preprint arXiv:2301.00182*, 2022.

[18] S. Li, H. Liu, R. Qian, Y. Li, J. See, M. Fei, X. Yu, and W. Lin, “Ta2n: two-stage action alignment network for few-shot action recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 1404–1411.

[19] L. Wu, Y. Wang, H. Yin, M. Wang, and L. Shao, “Few-shot deep adversarial learning for video-based person re-identification,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1233–1245, 2019.

[20] L. Wu, Y. Wang, J. Gao, and X. Li, “Where-and-when to look: Deep siamese attention networks for video-based person re-identification,” *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1412–1424, 2018.

[21] Z. Zhu, L. Wang, S. Guo, and G. Wu, “A closer look at few-shot video classification: A new baseline and benchmark,” *The British Machine Vision Conference*, 2021.

[22] T. Huang, B. Dong, Y. Yang, X. Huang, R. W. Lau, W. Ouyang, and W. Zuo, “Clip2point: Transfer clip to point cloud classification with image-depth pre-training,” *arXiv preprint arXiv:2210.01055*, 2022.

[23] H. Shi, M. Hayat, Y. Wu, and J. Cai, “Proposalclip: Unsupervised open-category object proposal generation via exploiting clip cues,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9611–9620.

[24] S. Esmailpour, B. Liu, E. Robertson, and L. Shu, “Zero-shot out-of-distribution detection based on the pretrained model clip,” in *Proceedings of the AAAI conference on artificial intelligence*, 2022.

[25] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, “Clip on wheels: Zero-shot object navigation as object localization and exploration,” *arXiv preprint arXiv:2203.10421*, 2022.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[27] L. A. Ramshaw and M. P. Marcus, “Text chunking using transformation-based learning,” in *Natural language processing using very large corpora*. Springer, 1999, pp. 157–176.

[28] L. Zhu and Y. Yang, “Label independent memory for semi-supervised few-shot video classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 273–285, 2020.

[29] A. Thatipelli, S. Narayan, S. Khan, R. M. Anwer, F. S. Khan, and B. Ghanem, “Spatio-temporal relation modeling for few-shot action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19958–19967.

[30] J. Wu, T. Zhang, Z. Zhang, F. Wu, and Y. Zhang, “Motion-modulated temporal fragment alignment network for few-shot action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9151–9160.

[31] X. Wang, S. Zhang, Z. Qing, M. Tang, Z. Zuo, C. Gao, R. Jin, and N. Sang, “Hybrid relation guided set matching for few-shot action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19948–19957.

[32] Y. Huang, L. Yang, and Y. Sato, “Compound prototype matching for few-shot action recognition,” in *European Conference on Computer Vision*. Springer, 2022, pp. 351–368.

[33] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, “The” something something” video database for learning and evaluating visual common sense,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5842–5850.

[34] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: a large video database for human motion recognition,” in *2011 International conference on computer vision*. IEEE, 2011, pp. 2556–2563.

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

[36] Y. Li, Y. Li, and N. Vasconcelos, “Resound: Towards action recognition without representation bias,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 513–528.

[37] D. Shao, Y. Zhao, B. Dai, and D. Lin, “Finegym: A hierarchical video dataset for fine-grained action understanding,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2616–2625.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[39] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, “Temporal segment networks for action recognition in videos,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2740–2755, 2018.

[40] N. Qian, “On the momentum term in gradient descent learning algorithms,” *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[41] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[42] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. 11, 2008.