# Temporal Relation based Attentive Prototype Network for Few-shot Action Recognition

**Guangge Wang** GUANGGEW@STU.XMU.EDU.CN
**Haihui Ye** HAIHUI_YE1@163.COM
**Xiao Wang** XIAOWANG@STU.XMU.EDU.CN
**Weirong Ye** WEIRONGYE@STU.XMU.EDU.CN
**Hanzi Wang*** WANG.HANZI@GMAIL.COM
*Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen, China*

**Editors:** Vineeth N Balasubramanian and Ivor Tsang

## Abstract

Few-shot action recognition aims at recognizing novel action classes with only a small number of labeled video samples. We propose a temporal relation based attentive prototype network (TRAPN) for few-shot action recognition. Concretely, we tackle this challenging task from three aspects. Firstly, we propose a spatio-temporal motion enhancement (STME) module to highlight object motions in videos. The STME module utilizes cues from content displacements in videos to enhance the features in the motion-related regions. Secondly, we learn the core common action transformations by our temporal relation (TR) module, which captures the temporal relations at short-term and long-term time scales. The learned temporal relations are encoded into descriptors to constitute sample-level features. The abstract action transformations are described by multiple groups of temporal relation descriptors. Thirdly, a vanilla prototype for the support class (e.g., the mean of the support class) cannot fit well for different query samples. We generate an attentive prototype constructed from temporal relation descriptors of support samples, which gives more weight to discriminative samples. We evaluate our TRAPN on Kinetics, UCF101 and HMDB51 real-world few-shot datasets. Results show that our network achieves the state-of-the-art performance.

**Keywords:** Few-shot Learning, Action Recognition, Temporal Relation Learning, Spatio-temporal Attention, Attentive Prototype

## 1. Introduction

Driven by the exponential growth of video data on the Internet, action recognition as a basic and core task of video understanding has developed rapidly. Many action recognition methods based on deep learning have achieved excellent performances ( Li et al. (2020); Zhonghong et al. (2020)). However, the successes are mainly due to the access to a large number of labeled videos. If there are not enough labeled videos, the model usually exhibits a decrease in accuracy because of insufficient training. In addition, humans need to consume much time and intensive labor to annotate extensive videos manually, which is impractical and expensive.

* Corresponding author.

Humans can learn unseen things from extremely constrained samples based on the past experience and knowledge. Motivated by this observation, the issue referred to as *few-shot learning* (FSL) ( Vinyals et al. (2016); Snell et al. (2017); Xiao et al. (2020)) that learning novel classes with few labeled samples has been greatly explored in recent years. In FSL, there is a base class set and a novel class set. The base class set does not share the common classes with the novel class set. FSL aims to learn the prior knowledge in the base class set and utilize the knowledge to recognize the novel classes with few labeled samples.

To achieve this, FSL is often elaborated as a meta-learning problem focusing on learning the prior knowledge across episode tasks ( Vinyals et al. (2016); Finn et al. (2017)). Specifically, each episode task is composed of a support set and a query set. The common learning scenario is to classify unseen samples from the query set into a set of new classes, given just a few labeled samples of each class from the support set. The key task of meta-learning is to train a base learner to map the embedding to task-relevant features with the goal of making the model generalize better to alleviate the difficulty caused by insufficient samples.

Classifying query samples by computing the distances between prototypes and the query samples ( Snell et al. (2017)), is a simple and efficient FSL method. Each prototype is represented by the mean value of the embedding sample features belonging to the same class from the support set. While this method is applied to few-shot action recognition ( Kumar Dwivedi et al. (2019); Fu et al. (2019, 2020)), there are still some limitations. Thus, we try to find better solutions to improve the prototype learning.

First, object motions are not highlighted. The discriminative information contained in the object motions is sometimes overwhelmed by static scene information. So similar scene appearance information in different classes may confuse the model and result in misclassifications. To tackle this problem, we propose a spatio-temporal motion enhancement module to enhance motion information modeling. The motion information is contained in the state changes of objects, which can be captured by computing the spatio-temporal content displacements in videos. Our module leverages the content displacements to produce motion weights, which are then utilized to enhance motion-related features. In this way, motion patterns are highlighted in the original features.

Second, the temporal features learned by data augmentation contain much noise. The synthesized data with the GAN ( Kumar Dwivedi et al. (2019)) and the shuffled temporal clips ( Fu et al. (2019, 2020)) may destroy the underlying structure of the action along the temporal dimension. The model tends to learn time-independent spatial local features instead of capturing action transformations. Inspired by ( Santoro et al. (2017)) and ( Zhou et al. (2018)), we focus on learning possible temporal relations across videos and discovering the core common properties of action transformations. Specifically, capturing temporal relations at a single time scale may be insufficient for the model to learn action features. Different paces, scene changes and appearance deformation make it hard even for humans to describe some actions at appropriate time scales. Thus, we propose the temporal relation module to capture temporal relations at short-term and long-term time scales, which alleviates the difficulty of learning the characteristics of abstract action transformations.

Third, samples of each class from the support set are directly averaged to generate vanilla prototypes. This means all of the samples from the support set are weighted equally. It turns out that the importances of different samples are not distinguished. Considering that the

dominant samples from the support set that are closer to the query samples are more likely to contain discriminative information, we propose the improved attentive prototype learning scheme. The expected prototypes for each class are obtained by attentive aggregation from the support samples, where the weights are computed by using the similarity scores for the corresponding query samples.

Based on the above mentioned, we propose a temporal relation based attentive prototype network (TRAPN) for few-shot action recognition. There are three main components in the proposed network: the spatio-temporal motion enhancement module, the temporal relation module and the attentive prototype metric. To be specific, we devise the spatio-temporal motion enhancement module to enhance motion information modeling, where motion patterns are highlighted across feature maps. Following motion enhancement, the temporal relation module sparsely samples video frames at different time scales to learn multiple groups of local temporal relation descriptors for each video. With the learned local temporal relation descriptors, the sample-level features can better preserve the captured temporal dynamics without losing considerable discriminative information. Finally, we leverage the learned temporal relation descriptors to create the attentive prototypes for action predictions. The network is optimized using the attentive prototype metric to minimize the distances between the query samples and the corresponding class prototypes.

In summary, the contributions of this paper are as follows:

- We propose the spatio-temporal motion enhancement module to highlight the motion-related features based on the motion attention mechanism.
- To make the model more effective in characterizing action representations, we design novel local temporal relation descriptors to efficiently capture action transformations of objects at multiple time scales.
- We propose an attentive prototype network, which can generate a high-quality prototype for each query sample. The experimental results verify the superiority of TRAPN over the state-of-the-art methods.

## 2. Related Work

**Action Recognition**. In recent years, convolution neural networks (CNNs) have been widely used in the video action recognition task. According to the convolutions used in the feature learning, these deep learning based methods can be briefly divided into two categories: 2D CNN based methods and 3D CNN based methods. 2D CNN based methods ( Wang et al. (2016); Zhou et al. (2018)) usually apply 2D CNNs to extract frame-level features of videos independently and then fuse extracted features along with the temporal dimension. However, spatio-temporal information in videos is not fully exploited. On the contrary, 3D CNN based methods ( Tran et al. (2015); Carreira and Zisserman (2017)) directly utilize spatio-temporal filters to learn motion features, which greatly increase both complexity and computational cost. Therefore, some methods ( Qiu et al. (2017); Tran et al. (2018)) attempt to combine the advantages of 2D CNNs and 3D CNNs to overcome the shortcomings of those methods.

**Few-shot Learning**. The few-shot learning aims to learn novel classes from very few labeled samples. To address the few-shot learning problem, some researchers focus on

learning powerful models with more generalization ability to better adapt to the novel classes based on the meta-learning strategy ( Vinyals et al. (2016)). The meta-learning based methods help to alleviate the overfitting problem to some extent. In addition, the metric learning based methods tackle the few-shot learning problem by learning a distance metric for samples. In the metric space, samples from the same class are close to each other but samples from different classes are far away. Specifically, the representative methods include Matching Network ( Vinyals et al. (2016)), Prototypical Network ( Snell et al. (2017)) and Relation Network ( Sung et al. (2018)).

**Few-shot Action Recognition**. Recent works have tackled the action recognition problem in the limited data scenario. CMN ( Zhu and Yang (2018)) proposes a memory network structure to store the feature representations. Embodied Learning ( Fu et al. (2019)) creates a virtual dataset to learn actions and leverages the proposed video segment augmentation method to synthesize new videos. ProtoGAN ( Kumar Dwivedi et al. (2019)) uses the Conditional GAN to synthesize video features for novel classes. AMeFu-Net ( Fu et al. (2020)) fuses the RGB modality and the depth modality to enhance the source video representations. ARN ( Zhang et al. (2020)) leverages the self-supervision data augmentation method to learn discriminative action features. In addition, TAM ( Cao et al. (2020)) proposes a temporal similarity metric method to dynamically align video sequences while learning temporal variations, and TARN ( Bishay et al. (2019)) aligns video sequences based on the attention mechanism. Compared with TAM and TARN, our method focuses more on learning general action features from the temporal sequence instead of treating few-shot action recognition as a video sequence matching problem.

## 3. Method

**Problem Definition**. In the few-shot action recognition problem, there is a meta-training dataset $\mathcal{D}_{train}$ and a meta-testing dataset $\mathcal{D}_{test}$. The classes of $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ are disjoint. An efficient strategy to solve the few-shot learning problem is to mimic the meta-learning setting via *episode* as proposed in ( Vinyals et al. (2016)). For each episode, $N$ different video classes $(\mathcal{C}_1, ..., \mathcal{C}_N)$ are randomly selected from the meta-training/meta-testing set and then $K$ labeled samples are sampled from each of the $N$ classes. Totally $N \times K$ samples constitute the *support set*. The *query set* is composed of one sample, which is sampled from the rest samples of the selected $N$ classes. Thus, an episode is also termed as a $N$-way $K$-shot task. The task is to classify the query video into one of the $N$ classes from the corresponding support set. The goal of our model is to quickly adapt to new tasks.

### 3.1. Pipeline

The overall structure of our model is shown in Figure 1. The input of our model consists of support videos and one query video. A raw video is usually composed of a sequence of frames. We adopt the sparse sampling strategy described in TSN ( Wang et al. (2016)) in order to avoid expensive computation of redundant content. All the sampled frames from each video are mapped to compact frame representations by the feature extractor ResNet ( He et al. (2016)). The spatio-temporal motion enhancement module is applied to the features extracted from ResNet, in order to highlight the features in the motion-related regions. Then the enhanced features are fed into the temporal relation module to

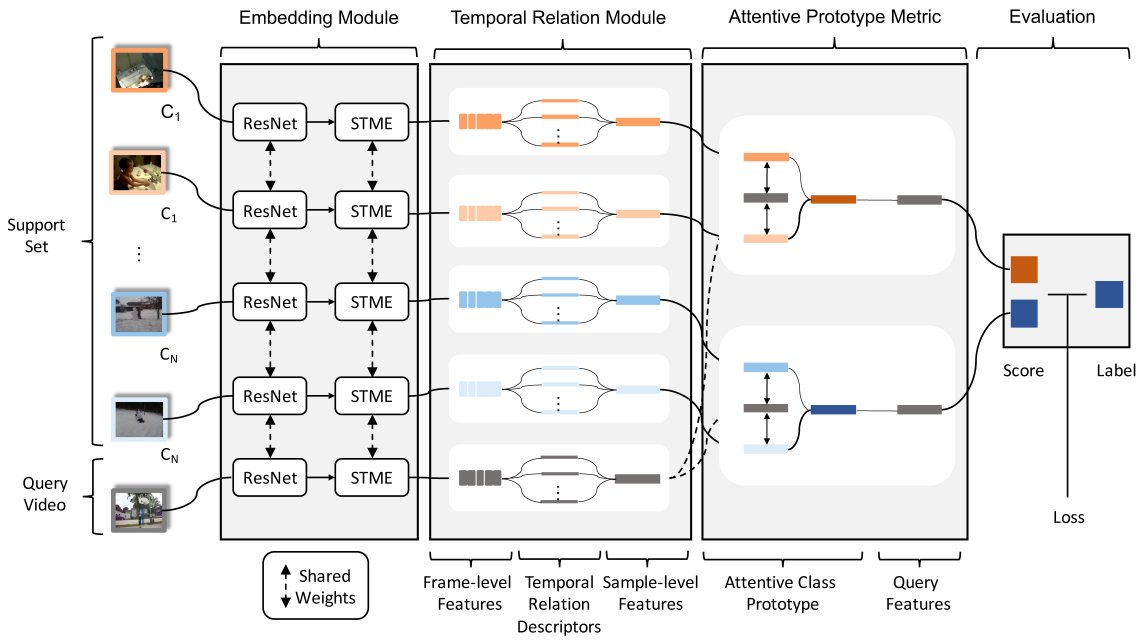


Figure 1: The pipeline of TRAPN. Each video input consists of the sampled frames from the video. First, the feature extractor ResNet processes each sampled frame individually and the spatio-temporal motion enhancement module highlights the motion-salient features. Then, we leverage the temporal relation module to learn temporal relation descriptors. Finally, we obtain the predicted action class of the query video by the attentive prototype metric.

learn temporal relation descriptors at multiple time scales. Finally, we get the video-to-class probability scores by computing the temporal relation similarities between the query sample and the attentive class prototypes by using the attentive prototype metric.

### 3.2. Spatio-temporal Motion Enhancement Module

Motion information plays a crucial role in understanding human behaviors in videos. Consequently, we propose a spatio-temporal motion enhancement (STME) module to focus on highlighting the motion-salient features. In fact, motion information can be measured by computing the content displacements of two successive frames ( Li et al. (2020)). Intuitively, the STME module enhances the features in motion-related regions based on a motion attention mechanism by utilizing cues from all the spatio-temporal content displacement positions.

We design the STME module following the non-local architecture ( Wang et al. (2018)). As shown in Figure 2, the input of STME is the spatio-temporal features $\boldsymbol{S}$ extracted from ResNet, where $\boldsymbol{S} \in \mathbb{R}^{T\times C\times H\times W}$. $T$ denotes the temporal dimension and $C$ denotes the feature channels. $H$ and $W$ denote the spatial size.

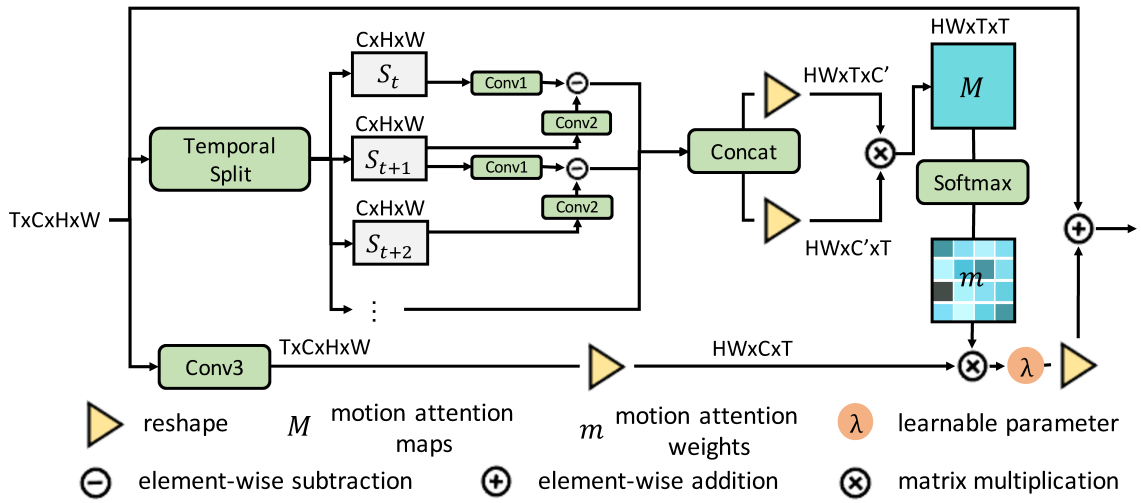


Figure 2: The architecture of the spatio-temporal motion enhancement (STME) module.

For calculating motion features, we first split the spatio-temporal features $\boldsymbol{S}$ into $T$ frame-level features along the temporal dimension. The motion features at the time step $t$ are approximately measured by the feature differences between adjacent frames, $\boldsymbol{S}_t$ and $\boldsymbol{S}_{t+1}$, where $t \in [1, T-1]$. For efficient computation, the $\boldsymbol{S}_t$ and $\boldsymbol{S}_{t+1}$ are respectively fed into two different $1 \times 1 \times 1$ convolutions, $conv_1$ and $conv_2$, by which the number of channels will be reduced. Formally,

$$\boldsymbol{D}(t) = conv_2(\boldsymbol{S}_{t+1}) - conv_1(\boldsymbol{S}_t), 1 \leq t \leq T-1, \tag{1}$$

where $\boldsymbol{D}(t) \in \mathbb{R}^{C' \times H \times W}$ are the motion features at the time step $t$ and $C'$ denotes feature channels after dimension reduction. In our experiments, $C'$ is set to $C/8$. To keep the temporal scale consistent with the input $\boldsymbol{S}$, we denote the motion feature at the time step $T$ as zero, i.e., $\boldsymbol{D}(T) = 0$. We concatenate all the motion features along the temporal dimension to construct the motion matrix $\boldsymbol{D} = [\boldsymbol{D}(1), ..., \boldsymbol{D}(T)]$, where $\boldsymbol{D} \in \mathbb{R}^{T \times C' \times H \times W}$.

Basically, we note that the motion features at the current time step can be further enhanced by considering the motion information at neighboring time steps. In practice, we first perform motion correlation computation between neighboring time instances to get the motion attention maps $\boldsymbol{M}$ and then get the motion attention weights $\boldsymbol{m}$. The calculation can be formulated as:

$$\boldsymbol{m}_{p,ji} = \frac{exp(\boldsymbol{M}_{p,ij})}{\sum_{i=1}^{T} exp(\boldsymbol{M}_{p,ij})}, where\ \boldsymbol{M}_{p,ij} = \boldsymbol{D}_{p,i}{}^{T} \boldsymbol{D}_{p,j}, \tag{2}$$

where $\boldsymbol{m}_{p,ji}$ indicates the extent to which the module attends to the $i^{th}$ time instances when synthesizing the $j^{th}$ time instances for the position at $p$.

Then, we apply motion attention weights $\boldsymbol{m}$ on $conv_3(\boldsymbol{S})$, which is the transformed features of $\boldsymbol{S}$ in the new feature space. Finally, we multiply the weighted features with a

learnable scalar parameter $\lambda$ and add back the input features $\boldsymbol{S}$ to get the output features $\boldsymbol{F}$. The calculation can be formulated as:

$$\boldsymbol{F}_{p,j} = \lambda \sum_{i=1}^{T} \boldsymbol{m}_{p,ji} conv_3(\boldsymbol{S}_{p,i}) + \boldsymbol{S}_{p,j}, \tag{3}$$

where $\boldsymbol{F} \in \mathbb{R}^{T \times C \times H \times W}$ is the final enhanced features.

### 3.3. Temporal Relation Module

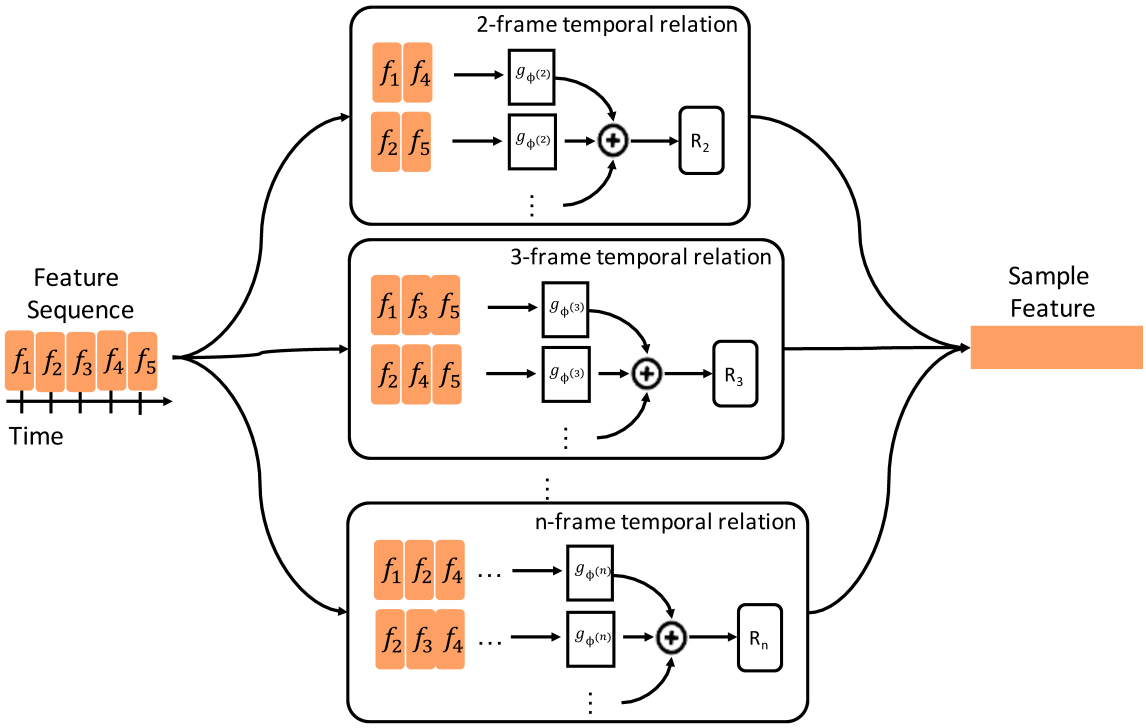


Figure 3: The illustration of the temporal relation (TR) module. TR samples different subsequences from the time-ordered feature sequence to learn multiple groups of temporal relation descriptors, where $R_n$ corresponds to the $n$-frame temporal relation descriptor. The learned temporal relation descriptors jointly constitute the sample features.

Inspired by the fact that humans can recognize action classes based on the observations of behavior changes across time, we propose the Temporal Relation (TR) module to mine the temporal dynamic information, which is captured by multiple groups of temporal relation descriptors at different time scales, as shown in Figure 3.

Specifically, we get the enhanced spatio-temporal features $\boldsymbol{F}$ after STME. We split $\boldsymbol{F}$ into $T$ ordered frame-level features $\boldsymbol{F} = \{f_1, f_2, ..., f_T\}$ along the temporal dimension. Since

the temporal relation should be captured from at least two frames, the 2-frame temporal relation descriptor can be defined as:

$$R_2(\boldsymbol{F}) = \sum_{a<b} g_{\phi^{(2)}}(f_a, f_b), \tag{4}$$

where $f_a$ is the $a^{th}$ frame feature and $f_b$ is the $b^{th}$ frame feature. $\{f_a, f_b\}$ is the possible subsequence of $\boldsymbol{F}$. The role of temporal relation function $g_{\phi^{(2)}}$ is to learn relations between frame features. $g_{\phi^{(2)}}$ is a fully connected layer with the parameters $\phi^{(2)}$. Since we aim to learn the common properties of the action transformation instead of any particular temporal relation, we sample multiple possible subsequences from $\boldsymbol{F}$ and accumulate the learned temporal relations.

Similarly, the 3-frame temporal relation descriptor can be defined as:

$$R_3(\boldsymbol{F}) = \sum_{a<b<c} g_{\phi^{(3)}}(f_a, f_b, f_c), \tag{5}$$

where $f_c$ is the $c^{th}$ frame feature. To capture temporal relations at multiple time scales, the temporal relation descriptor can be further extended as:

$$R_n(\boldsymbol{F}) = \sum_{a<b<c<n} g_{\phi^{(n)}}(f_a, f_b, f_c, ..., f_n), \tag{6}$$

where $R_n$ is the $n$-frame temporal relation descriptor, $n \leqslant T$. .

We randomly sample 3 subsequences from all possible subsequences for each temporal relation descriptor. Each temporal relation descriptor $R_n$ is set to be an $L$ dimensional vector. We combine all of these temporal relation descriptors together into a single tensor $\boldsymbol{X}$, $\boldsymbol{X} = [R_2, R_3, ..., R_n]$. Here, the sample feature $\boldsymbol{X}$ is obtained by stacking all the temporal relation descriptors, where $\boldsymbol{X} \in \mathbb{R}^{(T-1)\times L}$. In this way, the temporal dynamics at short-term and long-term time scales are explicitly encoded and the final sample-level features consist of the encoded descriptors. The effectiveness is validated in the ablation study (see in Figure 4).

### 3.4. Attentive Prototype Metric

Actually, not all the video samples in the same class are equally discriminative. Therefore, instead of directly fusing sample features of the same class averagely ( Snell et al. (2017)), we propose to generate attentive prototypes of support classes for each query sample.

Particularly, there are $N$ classes and each class has $K$ samples in a support set. $\boldsymbol{x_{ij}}$ is the $j^{th}$ sample of the $i^{th}$ class. And the query set has a query sample $\boldsymbol{q}$. Each sample is composed of multiple temporal relation descriptors obtained in Section 3.3, where $\boldsymbol{x_{ij}} = [x_{ij}^2, x_{ij}^3, ..., x_{ij}^n]$ and $\boldsymbol{q} = [q^2, q^3, ..., q^n]$. We define that the discriminability of each support sample is evaluated by the similarity with the query sample at the descriptor level. The discriminability value $\gamma_{ij}^n$ for the $n$-frame temporal relation descriptor $x_{ij}^n$ of the sample $\boldsymbol{x_{ij}}$ is calculated as:

$$\gamma_{ij}^n = \frac{exp(g(q^n, x_{ij}^n))}{\sum_{j=1}^{K} exp(g(q^n, x_{ij}^n))}, \tag{7}$$

where $g$ is the similarity function.

Then, we calculate the weighted $n$-frame temporal relation descriptor $p_i^n$ for the $i^{th}$ class:

$$p_i^n = \sum_{j=1}^{K} \gamma_{ij}^n x_{ij}^n, \tag{8}$$

where the discriminability value $\gamma_{ij}^n$ is the weight of $x_{ij}^n$. After generating each weighted temporal relation descriptor for the $i^{th}$ class, the attentive prototype $\boldsymbol{p_i} = [p_i^2, p_i^3, ..., p_i^n]$ for $\boldsymbol{q}$ is generated.

Finally, we obtain the predicted class scores for the query sample $\boldsymbol{q}$ by comparing the similarities between the query sample and different class prototypes,

$$P(i_{pre} = i|\boldsymbol{q}) = \frac{exp(\sum_{n=2}^{T} g(q^n, p_i^n))}{\sum_{i=1}^{N} exp(\sum_{n=2}^{T} g(q^n, p_i^n))}. \tag{9}$$

Particularly, each support prototype and the query sample have their own multiple groups of descriptors. We use the accumulated descriptor similarities between $\boldsymbol{q}$ and $\boldsymbol{p_i}$ as the video-to-class similarities. The softmax is appled over the video-to-class similarities to get the scores of predicted results.

## 4. Experiments

| Method | 1-shot | 2-shot | 3-shot | 4-shot | 5-shot |
|---|---|---|---|---|---|
| BaseNet | 66.4 | 76.2 | 79.9 | 81.6 | 83.3 |
| Matching Net ( Vinyals et al. (2016)) | 53.3 | - | - | - | 74.6 |
| MAML ( Finn et al. (2017)) | 54.2 | - | - | - | 75.3 |
| CMN ( Zhu and Yang (2018)) | 60.5 | 70.0 | 75.6 | 77.3 | 78.9 |
| TARN ( Bishay et al. (2019)) | 66.6 | 74.6 | 77.3 | 78.9 | 80.7 |
| CFA ( Hu et al. (2019)) | 69.9 | - | 80.5 | - | 83.1 |
| Embodied Learning ( Fu et al. (2019)) | 67.8 | 77.8 | 81.1 | 82.6 | 85.0 |
| ARN ( Zhang et al. (2020)) | 63.7 | - | - | - | 82.4 |
| TAM ( Cao et al. (2020)) | 73.0 | - | - | - | 85.8 |
| AMeFu-Net ( Fu et al. (2020)) | 74.1 | 81.1 | 84.3 | 85.6 | 86.8 |
| TRAPN (Ours) | **75.1** | **82.2** | **84.8** | **86.1** | **87.0** |

Table 1: Comparisons with the state-of-the-art methods on the Kinetics dataset. We report the 5-way action recognition accuracy (%) obtained on the meta-testing set.

### 4.1. Experiment Settings

**Datasets**. The Kinetics ( Carreira and Zisserman (2017)), UCF101 ( Soomro et al. (2012)) and HMDB51 ( Kuehne et al. (2011)) datasets have been frequently used to evaluate conventional action recognition in prior studies. The original Kinetics dataset has 306,245 videos

| Method | UCF101 | | | HMDB51 | | |
|---|---|---|---|---|---|---|
| | 1-shot | 3-shot | 5-shot | 1-shot | 3-shot | 5-shot |
| BaseNet | 78.6 | 90.2 | 92.7 | 48.9 | 62.4 | 67.9 |
| ProtoGAN ( Kumar Dwivedi et al. (2019)) | 62.3 | 75.6 | 80.5 | 35.7 | 46.6 | 51.5 |
| ARN ( Zhang et al. (2020)) | 66.3 | - | 83.1 | 45.5 | - | 60.6 |
| AMeFu-Net ( Fu et al. (2020)) | 85.1 | 93.1 | 95.5 | 60.2 | 71.5 | 75.5 |
| TRAPN (Ours) | **86.6** | **93.4** | **95.9** | **61.3** | **72.9** | **76.8** |

Table 2: Comparisons with the state-of-the-art methods on the UCF101 and HMDB51 datasets. We report 5-way action recognition accuracy (%) on the meta-testing sets.

and 400 action classes. UCF101 has 13,320 videos and 101 action classes. HMDB51 has 6,849 videos and 51 action classes. Particularly, we have to construct the few-shot versions to serve for few-shot action recognition. Following the dataset split strategy proposed in CMN ( Zhu and Yang (2018)), for the Kinetics, we sample 100 action classes from the original 400 action classes to construct the Kinetics subset. Specifically, 64, 12 and 24 non-overlapping classes are respectively constructed as the meta-training set, the meta-validation set and the meta-testing set. And each class has 100 videos. For UCF101 and HMDB51, we follow the same split strategy proposed in ( Zhang et al. (2020)). On UCF101, we respectively sample 31, 10 and 10 action classes as the meta-training set, the meta-validation set and the meta-testing set. And on HMDB51, we respectively sample 70, 10 and 21 action classes as the meta-training set, the meta-validation set and the meta-testing set.

**Implementation details**. We follow the sparse sampling strategy and the video pre-processing procedure proposed in TSN ( Wang et al. (2016)). Each input video sequence is divided into $T$ segments and one frame is randomly sampled from each segment. Subsequently, each frame is re-scaled to $256 \times 256$. All frames are augmented with random horizontal flips and then are cropped to obtain $224 \times 224$ regions.

In our experiment, we choose ResNet-50 pre-trained on ImageNet ( Deng et al. (2009)) as our backbone. The backbone is finetuned on the training data for 6 epoches, where the backbone with a learning rate of $1 \times 10^{-4}$. During the meta-training phase, we randomly select 2,000 episode tasks and choose Stochastic Gradient Descent (SGD) with momentum=0.9 to optimize the model parameters. For Kinetics, we set the learning rate to $2 \times 10^{-5}$. Considering that the scale of the UCF101 and HMDB51 datasets is relatively small, we set the learning rate to $1 \times 10^{-5}$. We tune the hyperparameters on the meta-validation set and stop the meta-training process until the accuracy on the meta-validation set begins to decrease. We evaluate our method on the standard $N$-way $K$-shot benchmark. The mean accuracy is calculated by randomly selecting 10,000 episodes from the meta-testing set in all experiments.

### 4.2. Comparison with State-of-the-Arts

In order to evaluate the performance of our model, we first compare our model with several state-of-the-art few-shot action recognition methods on the Kinetics, UCF101 and HMDB51

datasets. On the three datasets, we conduct experiments under 5-way $K$-shot settings. All methods use the same meta-training/testing set split.

**Baseline**. For the BaseNet baseline, we follow the setting of "BaseNet+test" proposed in Embodied Learning ( Fu et al. (2019)). We directly use the ResNet-50 pre-trained on ImageNet ( Deng et al. (2009)) as our backbone. BaseNet sparsely samples 8 frames from each video and then averages frame-level features along the temporal dimension to get video-level features. Specially, BaseNet uses ProtoNet ( Snell et al. (2017)) to get the trainable prototypes for each class. We use the cosine distance, instead of the Euclidean distance, for the similarity computation. The baseline and their settings are the same on Kinetics, UCF101 and HMDB51.

**Results on Kinetics**. Table 1 provides the comparative results obtained by the competing methods and our model on Kinetics. Our model significantly outperforms the baseline and all competing methods under different shot settings. We first note that the results of the baseline method BaseNet are relatively competitive compared with the other methods, and it even outperforms CMN ( Zhu and Yang (2018)) and ARN ( Zhang et al. (2020)) under the 1-shot setting. This demonstrates the superiority of BaseNet. We conclude that with the proper frame sampling protocol and training strategy, a model can be trained to generalize well to the unseen meta-testing set. Additionally, it can be seen that the proposed TRAPN significantly improves the performance of the baseline under all shot settings, especially with an accuracy increase of 8.7% under the 1-shot setting. And the proposed TRAPN achieves the state-of-the-art results with 75.1%, 82.2%, 84.8%, 86.1% and 87.0%, under the 1-shot, 2-shot, 3-shot, 4-shot and 5-shot settings, respectively. Specifically, ARN and AMeFu-Net ( Fu et al. (2020)) respectively use 20 frames and 16 frames as input, while our TRAPN just uses 8 frames as input. It shows that the temporal relation extraction contributes to the abstract action transformation learning to efficiently improve the few-shot action recognition performance even with fewer frames. The improvements are better when the labeled samples are extremely scarce, especially when there is only one labeled sample. Thus, our TRAPN is more prominent when there are fewer samples.

**Results on UCF101 and HMDB51**. The superiority of our TRAPN on UCF101 and HMDB51 is also impressive. We report the 1-shot, 3-shot and 5-shot accuracies in Table 2. The strong baseline BaseNet also shows the outstanding performance. Our TRAPN performs similar improvements with the results on Kinetics. It verifies the strong generalization ability of TRAPN on different datasets. Specifically, for UCF101, we achieve 86.6% under the 1-shot setting, 93.4% under the 3-shot setting and 95.9% under the 5-shot setting. For HMDB51, our method achieves 61.3% under the 1-shot setting, 72.9% under the 3-shot setting and 76.8% under the 5-shot setting.

Our TRAPN explicitly mines temporal dynamic features in videos and sufficiently leverages them to learn class prototypes. Our TRAPN achieves new state-of-the-art results on different datasets.

### 4.3. Ablation Study

As aforementioned, our method mainly benefits from the components contained in our model. We conduct ablation studies to study how those components contribute to our model. Generally, our ablation studies are conducted under the 5-way setting. We report

| Method | base | STME | TR | APM | 1-shot | 3-shot | 5-shot |
|---|---|---|---|---|---|---|---|
| (a) | ✓ | | | | 66.4 | 79.9 | 83.3 |
| (b) | ✓ | ✓ | | | 73.5 | 83.8 | 86.2 |
| (c) | ✓ | | ✓ | | 74.6 | 84.2 | 86.5 |
| (d) | ✓ | ✓ | ✓ | | 75.1 | 84.6 | 86.7 |
| (e) | ✓ | ✓ | ✓ | ✓ | **75.1** | **84.8** | **87.0** |

Table 3: Ablation studies on the spatio-temporal motion enhancement (STME) module, the temporal relation (TR) module and the attentive prototype metric (APM). The results are on Kinetics under the 5-way setting.

| Metric | 1-shot | 3-shot | 5-shot |
|---|---|---|---|
| Gaussian | 75.1 | 84.6 | 86.9 |
| Cosine | **75.1** | **84.8** | **87.0** |

Table 4: The performance with different similarity functions for the attentive prototype metric on Kinetics.

the results under the 1-shot, 3-shot and 5-shot settings in Table 3. We also compare the influence of the number of subsequences and the number of descriptors for TR in Figure 4.

**Analysis of of each component.** In this part, we report the individual influence of each component under different shot settings. As can be seen in Table 3, both STME and TR significantly improve the baseline by a large margin under the 1-shot setting. This shows that both STME and TR are beneficial for TRAPN to achieve the better action recognition performance.

Specifically, we use Grad-CAM ( Selvaraju et al. (2017)) to visualize the Class Activation Map (CAM) of some samples from Kinetics. The visualization results are shown in Figure 5. It can be seen that STME forces BaseNet to focus on the regions that are closely related to the action objects.

We also show that the performance can be further improved by aggregating STME and TR. The superior performance verifies that STME and TR are complementary in temporal motion modeling. In particular, we can observe that the improvements brought by STME and TR decrease as the number of shots increases. APM focuses more on discriminative samples at the descriptor level. The generated attentive prototypes are beneficial for the model to achieve the best results under the 3-shot and 5-shot settings.

**Analysis of the design of the temporal relation module.** The temporal relation module learns multiple groups of local temporal relation descriptors. The number of subsequences to learn temporal relation descriptors and the number of descriptors to constitute sample-level features are two parameters in the temporal relation module. How to

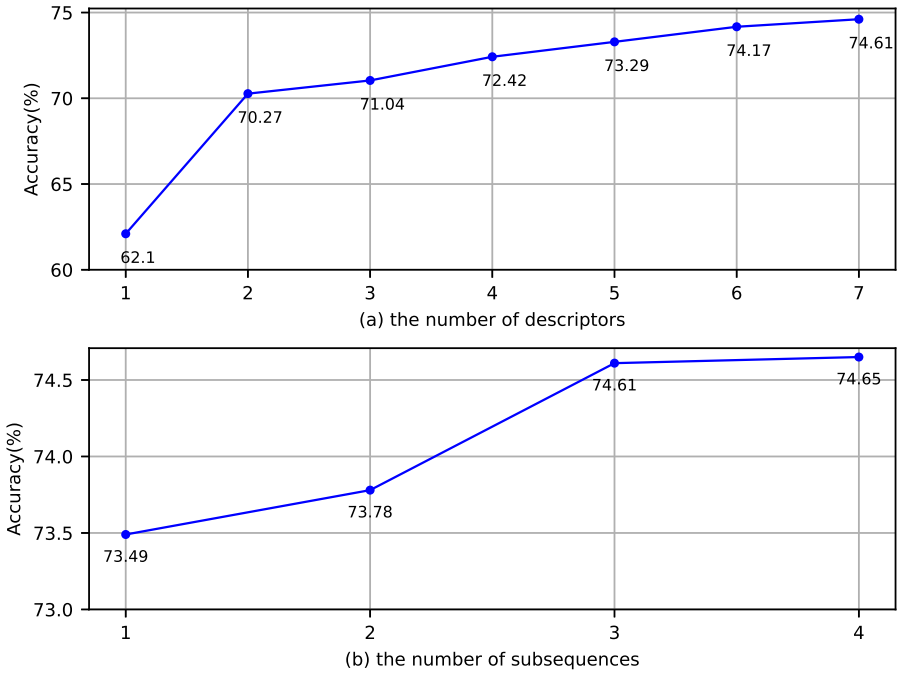


Figure 4: Influence of the number of subsequences and the number of descriptors on TR under the 1-shot setting.

choose the suitable parameter value is important to TR. As described above, $T$ frames are sampled from each video. We set the number of descriptors to 1 when the sample-level features are only composed of $[R_2]$. Therefore, the number of descriptors is $T-1$ when the sample-level features are composed of $[R_2, R_3, ..., R_T]$. Since $T$ is set to 8, the maximum number of descriptors is 7. All experiments perform similarly under the 1-shot setting. In Figure 4, the results demonstrate the significant improvement on Kinetics as the number of descriptors increasing from 1 to 7. As we can see, TR with 7 groups of descriptors achieves the highest accuracy of 74.61%. In addition, TR clearly benefits from selecting more subsequences. However, the performance of TR is saturated when the number of subsequences is more than 3. Thus, we sample 3 subsequences for each group of descriptors and use all the learned descriptors to construct the sample-level features in TR for all the experiments.

**Analysis of the similarity function $g$ for the attentive prototype metric.** In APM, we use a similarity function $g$ to measure the discriminability of sample descriptors. The choice of $g$ may affect the performance of APM. We choose the Gaussian similarity function and the cosine similarity function, which are two common metrics, for comparison. The results are shown in Table 4. It can be seen that the cosine similarity function performs better than the Gaussian similarity function. Thus, we adopt it as the similarity function $g$ in APM.

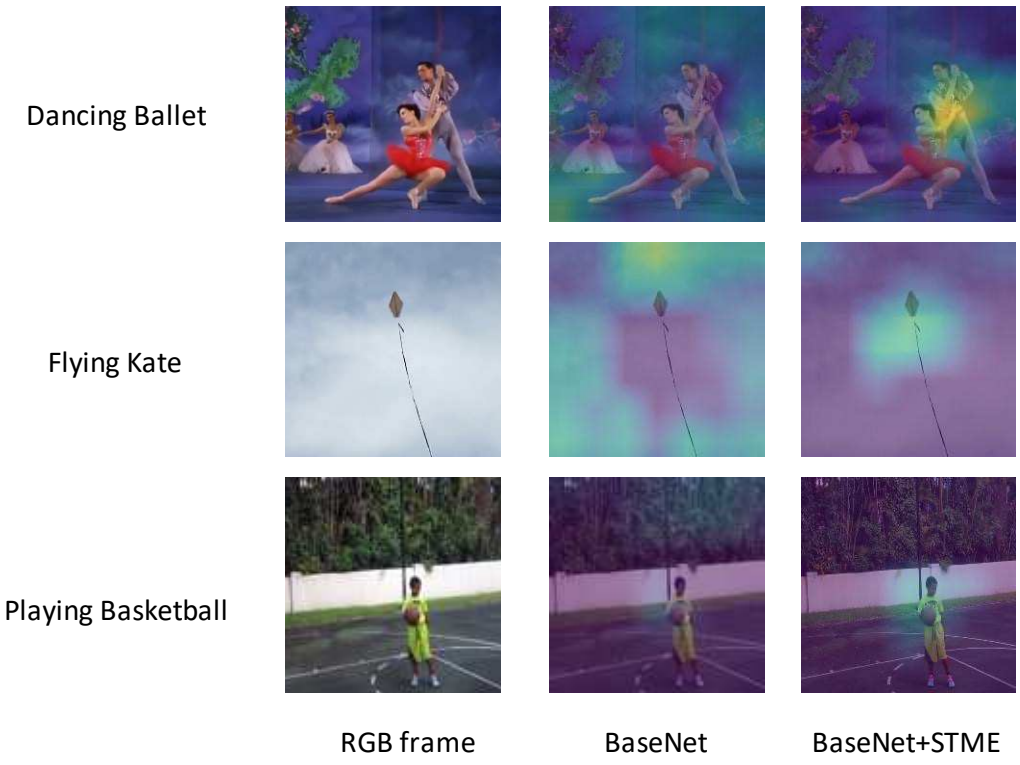


Figure 5: Visualization of the Class Activation Map (CAM) generated by our model. We show 3 samples from Kinetics. The first column shows the RGB frames. The CAM of BaseNet and BaseNet with STME are displayed in the second and third columns, respectively.

## 5. Conclusion

In this paper, we propose a temporal relation based attentive prototype network (TRAPN), including the components of spatio-temporal motion enhancement (STME) module, temporal relation (TR) module and attentive prototype metric, for few-shot action recognition. Specifically, the STME module can enhance spatio-temporal feature learning under the guide of motion information. Then, by considering the temporal dynamics contained in videos, we use the TR module to learn the temporal feature descriptors, whose importance is emphasized and verified. Furthermore, we focus more on discriminative samples of the same class at the descriptor level to measure the video-to-class similarities. Experimental results on three benchmark datasets demonstrate the effectiveness of our proposed TRAPN.

## References

Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv preprint arXiv:1907.09021*,

2019.

Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Computer Vision and Pattern Recognition*, pages 10618–10627, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

Yuqian Fu, Chengrong Wang, Yanwei Fu, Yu-Xiong Wang, Cong Bai, Xiangyang Xue, and Yu-Gang Jiang. Embodied one-shot video recognition: Learning from actions of a virtual embodied agent. In *ACM International Conference on Multimedia*, pages 411–419, 2019.

Yuqian Fu, Li Zhang, Junke Wang, Yanwei Fu, and Yu-Gang Jiang. Depth guided adaptive meta-fusion network for few-shot video recognition. In *ACM International Conference on Multimedia*, pages 1142–1151, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Ping Hu, Ximeng Sun, Kate Saenko, and Stan Sclaroff. Weakly-supervised compositional feature aggregation for few-shot recognition. *arXiv preprint arXiv:1906.04833*, 2019.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *International Conference on Computer Vision*, pages 2556–2563, 2011.

Sai Kumar Dwivedi, Vikram Gupta, Rahul Mitra, Shuaib Ahmed, and Arjun Jain. Protogan: Towards few shot learning for action recognition. In *International Conference on Computer Vision Workshops*, 2019.

Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Computer Vision and Pattern Recognition*, pages 909–918, 2020.

Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *International Conference on Computer Vision*, pages 5533–5541, 2017.

Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. 2017.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision*, pages 618–626, 2017.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *Neural Information Processing Systems*, 2017.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision*, pages 4489–4497, 2015.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Neural Information Processing Systems*, 2016.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36, 2016.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

Bin Xiao, Chien-Liang Liu, and Wen-Hoar Hsaio. Proxy network for few shot learning. In *Asian Conference on Machine Learning*, pages 657–672, 2020.

Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip HS Torr, and Piotr Koniusz. Few-shot action recognition with permutation-invariant attention. In *European Conference on Computer Vision*, 2020.

Li Zhonghong, Yi Yang, She Ying, Song Jialun, and Wu Yukun. Aarm: Action attention recalibration module for action recognition. In *Asian Conference on Machine Learning*, pages 97–112, 2020.

Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision*, pages 803–818, 2018.

Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *European Conference on Computer Vision*, pages 751–766, 2018.