# Compound Prototype Matching for Few-shot Action Recognition

Yifei Huang[0000−0001−8067−6227]⋆♯, Lijin Yang[0000−0003−3787−6658]⋆, and Yoichi Sato[0000−0003−0097−4537]

Institute of Industrial Science, the University of Tokyo, Tokyo, Japan
{hyf,yang-lj,ysato}@iis.u-tokyo.ac.jp

**Abstract.** Few-shot action recognition aims to recognize novel action classes using only a small number of labeled training samples. In this work, we propose a novel approach that first summarizes each video into compound prototypes consisting of a group of global prototypes and a group of focused prototypes, and then compares video similarity based on the prototypes. Each global prototype is encouraged to summarize a specific aspect from the entire video, *e.g.*, the start/evolution of the action. Since no clear annotation is provided for the global prototypes, we use a group of focused prototypes to focus on certain timestamps in the video. We compare video similarity by matching the compound prototypes between the support and query videos. The global prototypes are directly matched to compare videos from the same perspective, *e.g.*, to compare whether two actions start similarly. For the focused prototypes, since actions have various temporal variations in the videos, we apply bipartite matching to allow the comparison of actions with different temporal positions and shifts. Experiments demonstrate that our proposed method achieves state-of-the-art results on multiple benchmarks.

**Keywords:** Few-shot action recognition, compound prototype, prototype matching.

## 1 Introduction

Difficulty in collecting large-scale data and labels promotes the research on few-shot learning. Built upon the success of few-shot learning for image understanding tasks [33,10,62,26,16,54,36,37,53], many begin to focus on the few-shot action recognition task [28]. Once realized, such techniques could greatly alleviate the cost of video labeling [20] and promote real-world applications where the labels in certain scenarios are hard to acquire [25].

Most few-shot action recognition works determine the category of a query video using its similarity to the few labeled support videos. Many works [73,74,18] follow ProtoNet [47] to first learn a prototype for each video and compute the video similarity based on the similarity of the prototypes. To better consider the temporal dependencies in videos (*e.g.*, temporal ordering and relation), recent

---

⋆ Equal contribution. ♯ Corresponding author.

works construct sub-sequence prototypes using different parts of the videos and calculate video similarity by matching the support-query prototypes [5,44].

While temporal dependencies are considered, limitations still exist in previous methods. Firstly, without considering spatial information, previous methods cannot fully exploit the spatiotemporal relation in the videos for distinguishing actions like "put A on B" and "put A besides B" because they differ only in the relative position of the objects. Secondly, the sub-sequence prototypes come from fixed temporal locations, so that they cannot well handle the actions that happen at different speeds. Thirdly, it is computationally costly to exhaustively compute the similarity between all pairs of sub-sequence prototypes [44].

To address the limitations and achieve more robust few-shot action recognition, we explore how to: (1) better generate prototypes that can robustly encode spatiotemporal relation in the videos, (2) enable the prototypes to flexibly encode the actions done with different lengths and speeds, and (3) match the prototypes between two videos without exhaustively comparing all prototype pairs.

One straightforward way to address the first point is to use object features extracted from object bounding boxes [22]. However, we observed in our preliminary experiments only limited performance gain when previous methods [44,70] directly use them as additional inputs. In this work, we propose a multi-relation encoder to effectively encode the object features, by considering the spatiotemporal relation among objects across frames, the temporal relation between different frame-wise features, as well as the relation between object features and frame-wise features.

For the second and third points, we propose to generate *global prototypes* that consider all frames in the input video. This is done by taking advantage of the self-attention mechanism of Transformers [51]. Since it is difficult to represent a wide variety of actions by using a single prototype, we instead use a group of prototypes to represent each action. We match the support-query similarity by fixed 1-to-1 matching, *i.e.*, the $i$-th prototype of the query video is always matched with the $i$-th prototype of the support video. Thus, during training, each prototype will try to capture a certain aspect of the video. To avoid all prototypes to be the same, we apply a diversity loss when learning the prototypes, so that they are encouraged to capture different aspects of the action (*e.g.*, one prototype captures the start of the action, and another prototype captures the action evolution).

However, learning prototypes to represent different aspects of the action (*e.g.*, the start/end of the action) is a difficult task even with annotation [9,40,41]. Thus, it is not sufficient to compute video similarity only by the global prototypes. To make the similarity measurement more reliable, we generate another group of *focused prototypes*, where each prototype is encouraged to focus on certain timestamps of the video. Since actions may happen in different parts of the videos at different speeds [34], it is not correct to compare the same timestamp between two videos. We therefore use bipartite matching to match the focused prototypes between the support and query videos, so that comparison of actions with different lengths and speeds can be done.

Our method uses the compound of two groups of prototypes, which we call *compound prototypes*, for calculating video similarity. On multiple benchmark datasets, our method outperforms previous methods significantly when only one example is available, demonstrating the superiority of using our compound prototypes in similarity measurement.

To summarize, our key contributions include:

- A novel method for few-shot action recognition based on generating and matching compound prototypes.
- Our method achieves state-of-the-art performance on multiple benchmark datasets [7,19,30], outperforming previous methods by a large margin.
- A detailed ablation study showing the usefulness of leveraging object information for few-shot action recognition and demonstrating how the two groups of prototypes encode the video from complementary perspectives.

## 2   Related Works

**Few-shot image classification** methods can be broadly divided into three categories. The transfer-learning based methods [14,45,57] use pre-training and fine-tuning to increase the performance of deep backbone networks on few-shot learning. The second line of works focuses on rapidly learning an optimized classifier using limited training data [1,21,46,2,58,17,67]. The third direction is based on metric learning, whose goal is to learn more generalizable feature embeddings under a certain distance metric [29,47,52,27]. The key to metric-learning based methods is to generate robust representations of data under a certain metric, so that it may generalize to novel categories with few labeled samples. Our work falls into this school of research and focus on the more challenging video setting.

**Few-shot action recognition** methods [50,34,43,24,3,74,42,75,61] mainly fall into the metric-learning framework. Many works follow the scheme of ProtoNet [47] to compute video similarity based on generated prototypes. For learning better prototypes, ProtoGAN [32] synthesizes additional feature, CMN [72,73] uses a multi-layer memory network, while ARN [69] uses jigsaws for self-supervision and enhances the video-level representation via spatial and temporal attention. There are also methods that perform pretraining with semantic labels [56,59,60] or use additional information such as depth [18] to augment video-level prototypes. Our method uses another form of additional information: the object bounding box from pretrained object detectors [22].

The temporal variance of actions form a major challenge in few-shot action recognition task [34]. To better model temporal dependencies, recent researches put more focus on generating and matching sub-video level prototypes. OTAM [5] uses a generalized dynamic time warping technique [8] to monotonously match the prototypes between query and support videos. ITA-Net [70] first implicitly aggregates information for each frame using other frames, and conducts a 1-to-1 matching of all prototypes. However, these two methods use frame-wise prototypes, thus cannot well capture the higher-level temporal relation in multiple frames. Recently, TRX [44] constructs prototypes of different

cardinalities for query and support videos, and calculate similarity by matching all prototype pairs. However, since TRX can only match prototypes with the same cardinality (*e.g.*, three-frame prototypes matches with three-frame prototypes), it is hard to align the actions with different evolving speeds (*e.g.*, one takes 2 frames while the other takes 4 frames). Also, the exhaustively matching of all pairs is computationally expensive.

We summarize three main differences compared with previous works: (1) We encode spatiotemporal object information to form more robust prototypes. (2) We generate a compound of global and focused prototypes to represent the actions from diverse perspectives. (3) The two groups of prototypes are efficiently matched to robustly compute video similarity.

**Transformers** [51] have recently acquired remarkable achievements in computer vision [6,38,13,15,68,64]. FEAT [65] represents early work that applies transformer in the few-shot learning task, and TRX [44] first introduces Transformer [15] into the few-shot action recognition task. Different from TRX, we apply a Transformer encoder-decoder to generate compound prototypes, and show that this is more effective in the few-shot action recognition scenario.

## 3    Method

### 3.1    Problem Setting

In few-shot action recognition, a model aims to recognize an unlabeled video (query) into one of the target categories each with a limited number of labeled examples (support set) [5,44]. We follow the common practice [52,17] to use episodic training, where in each episode a $C$-way $K$-shot problem is sampled: the support set $\mathcal{S} = \{\boldsymbol{X}_s^j\}_{j=1}^{C \times K}$ is composed of $C \times K$ labeled videos from $C$ different classes where each class contains $K$ samples. The query set contains $N$ unlabeled videos $\mathcal{Q} = \{\boldsymbol{X}_q^i\}_{i=1}^N$. The goal is to classify each video in the query set as one of the $C$ classes.

### 3.2    Proposed Method

We propose a new method for few-shot action recognition by generating and matching compound prototypes between query and support videos. As shown in Fig. 1, given the video input, a feature embedding network first extracts global (frame-wise) features $\boldsymbol{F}_g$. To better model the actions involving multiple objects, we obtain object bounding boxes by a pretrained object detector [22], and extract object features $\boldsymbol{F}_o$ using the same embedding network. Then a multi-relation encoder uses $\boldsymbol{F}_g$ and $\boldsymbol{F}_o$ to output multi-relation features $\boldsymbol{F}_m$ containing spatiotemporal global-object relations. Then a compound prototype decoder generates global prototypes $\boldsymbol{P}_g$ and focused prototypes $\boldsymbol{P}_f$ for each video. During similarity calculation, we use fixed 1-to-1 matching on the global prototypes and bipartite matching on the focused prototypes, encouraging the similarity to be computed robustly from diverse perspectives. We introduce the details of each component below.
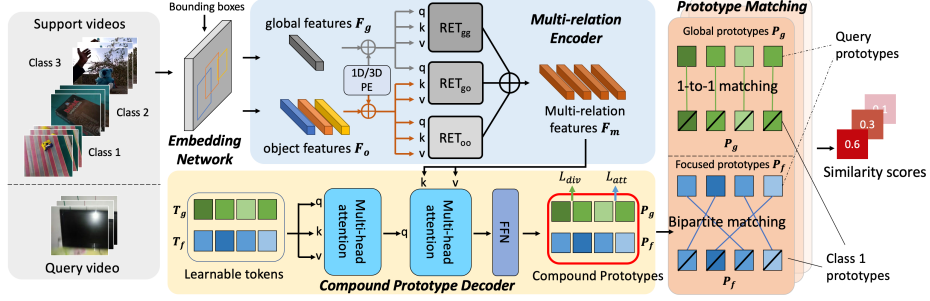
**Fig. 1.** Illustration of our proposed method on a 3-way 1-shot problem. First, the videos are processed by an embedding network to acquire global (frame-level) features $\boldsymbol{F}_g$ and object features $\boldsymbol{F}_o$. Features $\boldsymbol{F}_g$ and $\boldsymbol{F}_o$ are equipped with 1D and 3D positional encoding (PE), respectively, and then used by a multi-relation encoder (Sec. 3.2) to encode global-global, global-object and object-object information into a multi-relation feature $\boldsymbol{F}_m$. Using $\boldsymbol{F}_m$, a Transformer-based compound prototype decoder transforms the learnable tokens $\boldsymbol{T}_g, \boldsymbol{T}_f$ into compound prototypes that represent the input video (Sec. 3.2). The compound prototypes is consisted of several global prototypes $\boldsymbol{P}_g$ (green squares) and focused prototypes $\boldsymbol{P}_f$ (blue squares). They are applied with different loss and different matching strategies, so that each global prototype captures a certain aspect of the action summarized from the whole video, and each focused prototype focuses on a specific temporal location of the video. The final similarity score is calculated as the average similarity of all matched prototype pairs between support and query videos.

**Feature embedding** For each input video $\boldsymbol{X} \in \mathcal{S} \cup \mathcal{Q}$, we sample $T$ frames following the sampling strategy of TSN [55]. We use an embedding network to acquire a global (frame-level) feature representation for each video $\boldsymbol{F}_g \in \mathbb{R}^{T \times d}$, where $d$ is the feature dimension. Additionally we extract object features via ROI-Align [22] using the predicted bounding boxes on each frame. We use only $B$ most confident boxes on each frame, forming object features $\boldsymbol{F}_o \in \mathbb{R}^{BT \times d}$.

**Multi-relation encoder** To better generate prototypes that are discriminative for actions involving multiple objects, we propose to use a multi-relation encoder to encode the spatiotemporal information from $\boldsymbol{F}_g$ and $\boldsymbol{F}_o$. We specifically consider the following three relations: global-global (frame-wise) relation, global-object relation, object-object relation, and apply transformer [51] as the base architecture to allow relation modeling across frames. As shown in Fig. 1, the encoder consists of three relation encoding transformers (RETs). The global-global RET ($\text{RET}_{gg}$) and the object-object RET ($\text{RET}_{oo}$) are identical except the input. They take as input the global feature $\boldsymbol{F}_g$ and the object feature $\boldsymbol{F}_o$ respectively, and use the input to generate the query $\boldsymbol{Q}$, key $\boldsymbol{K}$ and value $\boldsymbol{V}$ vectors for the transformer:

$$\boldsymbol{F}_{gg} = RET_{gg}(\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{F}_g) \tag{1}$$

$$\boldsymbol{F}_{oo} = RET_{oo}(\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{F}_o), \tag{2}$$

The global-object RET ($RET_{go}$) works differently, where it maps $\boldsymbol{F}_g$ as query vector, while $\boldsymbol{F}_o$ as key and value vectors:

$$\boldsymbol{F}_{go} = RET_{go}(\boldsymbol{Q} = \boldsymbol{F}_g, \ \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{F}_o). \tag{3}$$

The output size of each RET is the same as its input query vector. Thus, each of the $T$ frame would have $B+2$ features with dimension $d$. We concatenate the three outputs into a multi-relation feature $\boldsymbol{F}_m \in \mathbb{R}^{(B+2)T \times d}$.

Positional encoding (PE) is shown to be effective in transformer-based architectures [51,39,38]. We also use PE but omit in the equations for simplicity. For $\boldsymbol{F}_g$ we use 1D PE to encode the temporal location of each frame, and for $\boldsymbol{F}_o$ we apply 3D PE, encoding both spatial and temporal location of each object.

**Compound prototype decoder** The compound prototype decoder also follows the transformer architecture [51,49,11,6], so that the prototypes can be generated by considering feature of all frames via self-attention. As shown in Fig. 1, the input to the prototype decoder are two groups of learnable tokens $\boldsymbol{T}_g \in \mathbb{R}^{m_g \times d}$ and $\boldsymbol{T}_f \in \mathbb{R}^{m_f \times d}$. A multi-head attention layer first encodes the tokens into $\hat{\boldsymbol{T}}_g$ and $\hat{\boldsymbol{T}}_f$, then another multi-head attention layer transforms them into two groups of prototypes $\boldsymbol{P}_g = \{\boldsymbol{p}_{g,k}\}_{k=1}^{m_g} \in \mathbb{R}^{m_g \times d}$ and $\boldsymbol{P}_f = \{\boldsymbol{p}_{f,k}\}_{k=1}^{m_f} \in \mathbb{R}^{m_f \times d}$. For simplicity, we omit the subscripts $_{g,f}$ and all normalization layers, thus the equation can be written as:

$$\boldsymbol{Q} = \hat{\boldsymbol{T}}\boldsymbol{W}_Q, \quad \boldsymbol{K} = \boldsymbol{F}_m\boldsymbol{W}_K, \quad \boldsymbol{V} = \boldsymbol{F}_m\boldsymbol{W}_V, \tag{4}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d \times d}$ are linear projection weights, then we have

$$\boldsymbol{A} = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}), \quad \boldsymbol{P} = FFN(\boldsymbol{A}\boldsymbol{V}), \tag{5}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times (B+2)T}$ is the self-attention weights, and FFN denotes feed forward network.

To encourage the prototypes to capture different aspects of the action, we apply constraints on the two groups of prototypes individually. For the global prototypes $\boldsymbol{P}_g$, we add a diversity loss to maximize their diversity:

$$L_{div} = \sum_{i \neq j} sim(\boldsymbol{p}_{g,i}, \ \boldsymbol{p}_{g,j}), \tag{6}$$

where $sim$ denotes the cosine similarity function.

Since learning $\boldsymbol{P}_g$ to robustly represent each aspect of the action (*e.g.*, the start of the action) is difficult even with annotation [63,66]. To increase the overall robustness, for the focused prototypes $\boldsymbol{P}_f$, we instead add regularization on the self-attention weight $\boldsymbol{A}_f$ so that different $\boldsymbol{p}_f$ can focus on different temporal locations of the video:

$$L_{att} = \sum_{i \neq j} sim(\boldsymbol{\alpha}_{f,i}, \ \boldsymbol{\alpha}_{f,j}). \tag{7}$$

Here $\boldsymbol{\alpha}_{f,i} \in \mathbb{R}^{(B+2)T}$ denotes the $i$-th row in $\boldsymbol{A}_f$.

**Compound Prototype Matching** Cooperating with the compound proto-types, we use different matching strategies to calculate the overall similarity between two videos. As shown in Fig. 1, for the global prototypes $\boldsymbol{P}_g$, we match the query and support prototypes in a 1-to-1 manner, *i.e.*, the $i$-th prototype of the query video is always matched with the $i$-th prototype of the support video. To calculate the global prototypes' overall similarity score between video $a$ and video $b$, we average the similarity score of all the $m_g$ global prototypes:

$$s_g^{a,b} = \frac{1}{m_g} \sum_{k=1}^{m_g} sim(\boldsymbol{p}_{g,k}^a, \ \boldsymbol{p}_{g,k}^b). \tag{8}$$

Thus, to maximize the similarity score of correct video pairs and minimize the similarity of incorrect video pairs during episodic training, each $\boldsymbol{p}_g$ will try to en-code a certain aspect of the action, *e.g.*, the start of the action. This phenomenon is supported by our experiments in Sec. 4.

For the focused prototypes $\boldsymbol{P}_f$, we apply a bipartite matching-based similar-ity measure. Since different actions may happen in different temporal positions in the videos, the bipartite matching enables the temporal alignment of actions, allowing the comparison between actions of different lengths and at different speeds. Formally speaking, for $\boldsymbol{P}_f^a$ of video $a$ and $\boldsymbol{P}_f^b$ of video $b$, we find a bi-partite matching between these two sets of prototypes by searching for the best permutation of $m_f$ elements with the highest cosine similarity using the Hun-garian algorithm [31]. Denote $\sigma$ as the best permutation, the similarity score of $\boldsymbol{P}_f^a$ and $\boldsymbol{P}_f^b$ is calculated by:

$$s_d^{i,j} = \frac{1}{m_f} \sum_{k=1}^{m_f} sim(\boldsymbol{p}_{f,k}^a, \ \boldsymbol{p}_{f,\sigma(k)}^b). \tag{9}$$

Finally, the similarity score is computed by a weighted average of $s_g$ and $s_f$: $s^{a,b} = \lambda_1 s_g^{a,b} + \lambda_2 s_f^{a,b}$. During training, this similarity score is directly regarded as logits for the cross-entropy loss $L_{ce}$. The total loss function is a weighted sum of three losses: $L = w_1 L_{ce} + w_2 L_{div} + w_3 L_{att}$. During inference, we assign the label of the query video as the label of the most similar video in the support set.

## 4 Experiments

We conduct experiments on four public datasets. **Kinetics** [7] and Something-something V2 (**SSv2**) [19] are two most frequently used benchmarks for few-shot action recognition. These two datasets are both splitted as 64/12/24 classes for train/val/test. For **SSv2**, we use both the split from CMN [72] (**SSv2°**) and the split from OTAM [5] (**SSv2♯**). Recently, Zhang *et al.* [69] proposed new splits for **HMDB** [30] and **UCF** [48] datasets. We also conduct experiments on these two datasets using the split from [69].

Since our method's performance is competitive in both standard 1-shot 5-way setting and 5-shot 5-way setting, in this section we only demonstrate 1-shot

results and place 5 shot results in the supplementary due to page limitation. Following previous works, we report the average result of 10,000 test episodes in the experiments.

**Baselines** We compare our method with recent works reporting state-of-the-art performance, including MatchNet [52], CMN [73], OTAM [5], TRN [71], ARN [69], TRX [44], ITA-Net [70]. Following [74], we also compare with the few-shot image classification model FEAT [65] which is also based on transformers. Specifically, since no previous works used object detector in few-shot action recognition, for a fair comparison with previous works, we conduct experiments in two settings: (1) we give baseline methods the same input (both $F_g$ and $F_o$) as our method and compare the performance. We denote the baselines as "Baseline+" in this setting. (2) We discard the object detector in our method and use only $F_g$ as input and $RET_{gg}$ as the encoder. We denote our method in this setting as "Ours-".

To enable previous methods to take object features as input, we choose to compare with methods TRX+, FEAT+, and ITA-Net+ because they also use transformer-based architectures like our method, thus no modification on the model architecture is needed. For completeness, we also compare MatchNet+ and TRN+ without transformer architecture. For these two methods, we re-shape the object features $F_o$ to $F'_o \in \mathbb{R}^{T \times Bd}$ as input. Since these two works are not designed to input the object features, we train an ensembled network, one with $F_g$ as input and the other with $F'_o$ as input. We use the public implementation of TRX and implement ITA-Net by ourselves. More details about baseline implementation can be found in the supplementary material. Recently, several works [56,4,74] pretrain the backbone on the meta-training set and found this pretraining to be useful for few-shot action recognition. To compare with the majority of prior works, we do not follow this setting in our experiments.

**Implementation details** We use ResNet-50 [23] pretrained on ImageNet [12] as the backbone of embedding network, and a fixed Mask-RCNN [22] trained on COCO dataset [35] is used as the bounding box extractor. We select $B = 3$ most confident object bounding boxes per frame. The pre-processing steps follow OTAM [5] where we also sample $T = 8$ frames with random crop during training and center crop during inference. The model and the backbone is optimized using SGD with an initial learning rate of 0.001 and decaying every 20 epochs by 0.1. The embedding network except its first BatchNorm layer is fine-tuned with 1/10 learning rate. The training is stopped when the loss on the validation set is greater than the average of the previous 5 epochs. Unless otherwise stated, we report result using $m_g = m_f = 8$, $\lambda_1 = \lambda_2 = 0.5$, $w_1 = 1, w_2 = w_3 = 0.1$.

## 4.1  Results

Tab. 1 shows result comparison with baseline methods. In the upper block, our model slightly outperforms previous works even without using the object detector on 4 of the 5 dataset/splits. This suggests that compared with other

**Table 1.** Results of 5-way 1-shot experiments on 5 dataset splits. Methods marked with * indicates results of our implementation with the original reported results shown in parenthesis. The bottom and upper block are results with and without using object features, respectively. For ITA-Net, result on SSv2° comes from Table 5 of [70].

| Method | SSv2° | SSv2$^\sharp$ | Kinetics | HMDB | UCF |
|---|---|---|---|---|---|
| MatchNet* [52] | 34.9 (31.3) | 35.1 | 54.6 (53.3) | 50.1 | 70.3 |
| CMN [73] | 36.2 | - | 60.5 | - | - |
| ARN [69] | - | - | 63.7 | 45.5 | 66.3 |
| OTAM [5] | - | 42.8 | 73.0 | - | - |
| TRN* [71,5] | 35.9 | 39.6 (38.6) | 68.6 (68.4) | 52.3 | 76.3 |
| FEAT* [65] | 38.4 | 45.5 | 73.0 | 56.1 | 75.8 |
| ITA-Net* [70] | 38.4 (38.6) | 46.1 | 72.6 | 56.5 | 76.0 |
| TRX* [44] | 37.1 (36.0) | 41.5 (42.0) | 64.6 (63.6) | 54.4 | **77.7** |
| Ours- | **38.9** | **49.3** | **73.3** | **60.1** | 71.4 |
| MatchNet+* | 35.6 | 36.5 | 57.6 | 52.8 | 72.8 |
| TRN+* | 38.1 | 41.3 | 71.4 | 55.3 | 79.7 |
| FEAT+* | 43.1 | 46.6 | 73.9 | 61.5 | 79.7 |
| ITA-Net+* | 43.7 | 48.9 | 74.4 | 61.6 | 79.5 |
| TRX+* | 39.4 | 44.2 | 71.8 | 60.1 | **81.2** |
| Ours | **57.1** | **59.6** | **81.0** | **80.3** | 79.0 |

prototype generation methods, our proposed approach to generate and match the compound prototypes enables better similarity measurement for few-shot action recognition. From the comparison of baseline methods with their "+" variants, we can see that these methods cannot fully exploit the information brought by the object features. When object features are added as input, our full model significantly outperforms previous methods that use the same input. Compared with the performance of our method in the upper block, the object features significantly stimulates the potential of our proposed compound prototype matching scheme, by improving the accuracy on the SSv2 dataset for over 10%, the Kinetics dataset for 7.7% and the HMDB dataset for over 20%. In the ablation study, we show even better performance can be achieved by carefully adjusting the number of prototypes $m_g$ and $m_f$.

Our method does not achieve state-of-the-art results on the UCF dataset. One reason is that classes in the UCF dataset can be easily distinguished only from appearance. This causes our multi-relation encoder to overfit. If we remove this encoder and directly use concatenation of object and global features as input to the compound prototype decoder, our method can achieve the state-of-the-art accuracy of 87.7. This reveals one limitation of our method, *i.e.*, easy to overfit on simple datasets.

## 4.2 Ablation Study

**Effect of object features** While there exist previous studies that leverage additional information for few-shot action recognition [18], no work has inves-

**Table 2.** Results comparison of different methods when using/not using object information, and using/not using our proposed encoder.

| Encoding | Prototype | SSv2° | SSv2$^\sharp$ | Kinetics |
|---|---|---|---|---|
| Backbone only (w/o. object features) | None | 34.9 | 35.1 | 54.6 |
| | ITA-Net | **38.4** | 45.5 | **72.6** |
| | TRX | 37.1 | 41.5 | 64.6 |
| | Ours | 37.9 | **48.5** | 72.5 |
| Our encoder (w/o. object features) | None | 35.3 | 37.2 | 59.4 |
| | ITA-Net | 38.6 | 46.5 | 73.0 |
| | TRX | 38.2 | 44.4 | 68.9 |
| | Ours | **38.9** | **49.3** | **73.3** |
| Concat (w. object features) | None | 35.6 | 36.5 | 57.6 |
| | ITA-Net+ | **43.7** | 48.9 | 74.4 |
| | TRX+ | 39.4 | 44.2 | 71.8 |
| | Ours | 42.1 | **49.1** | **79.2** |
| Our encoder (w. object features) | None | 39.1 | 42.5 | 58.4 |
| | ITA-Net+ | 44.0 | 50.7 | 73.3 |
| | TRX+ | 46.3 | 48.4 | 73.4 |
| | Ours | **57.1** | **59.6** | **81.0** |

**Table 3.** Results comparison of our model using different encoding relations with different number of global prototypes and focused prototypes.

| Setting | $\boldsymbol{F}_{gg}$ | $\boldsymbol{F}_{go}$ | $\boldsymbol{F}_{oo}$ | SSv2$^\sharp$ | Kinetics |
|---|---|---|---|---|---|
| $m_g = 8,$ $m_f = 8$ | ✓ | × | × | 49.3 | 73.3 |
| | × | ✓ | × | 47.3 | 69.1 |
| | × | × | ✓ | 55.5 | 73.6 |
| | ✓ | ✓ | × | 49.6 | 75.0 |
| | ✓ | × | ✓ | 57.7 | 79.7 |
| | × | ✓ | ✓ | 57.8 | 77.1 |
| | ✓ | ✓ | ✓ | **59.6** | **81.0** |
| $m_g = 16,$ $m_f = 0$ | ✓ | × | × | 38.9 | 65.5 |
| | × | ✓ | × | 38.8 | 65.2 |
| | × | × | ✓ | 38.5 | 64.0 |
| | ✓ | ✓ | ✓ | **39.3** | **66.1** |
| $m_g = 0,$ $m_f = 16$ | ✓ | × | × | 41.9 | 72.3 |
| | × | ✓ | × | 34.4 | 73.7 |
| | × | × | ✓ | 29.3 | 61.0 |
| | ✓ | ✓ | ✓ | **44.0** | **76.7** |

tigated the use of object features as in our method. One may argue that the performance improvement of our method only comes from the use of object features. However, in Tab. 1 we find that performance gain of previous methods is limited if object features are additionally used as input. Here, we show that a boost in performance only happens when object features, our multi-relation encoder, and our decoder are used together.

We test the performance of multiple methods with and without using object information, and also with and without using our multi-relation encoder. In Tab. 2, the first block uses neither object feature nor our multi-relation encoder, and the second block uses our encoder but with only the global-global relation $RET_{gg}$. From the comparison between these two blocks, we can see that our multi-relation encoder can improve the performance of all methods, but not significantly. In the third block of Tab. 2, we concatenate each frame-wise feature with its corresponding object features as input. The comparison between this block and the first block shows the improvement brought by object features: around $2 \sim 5$ percent on SSv2 dataset, and $2 \sim 6$ percent on Kinetics dataset. Finally, in the fourth block of Tab. 2, both object features and our multi-relation encoder are used. Comparing this block with the second and third blocks, all methods get more improvement. This shows that using our multi-relation encoder to consider multiple relations across frames can better leverage the information brought by the object features. Among all methods in the fourth block of Tab. 2, our method enjoys the most significant performance gain. This indicates that while the object features bring additional information, our method can best leverage this information to help the few-shot action recognition task.

**Impact of multi-relation feature encoding** To see how does the multi-relation feature encoding contribute to the performance, we conduct ablation study of our method considering only subsets of $\{\boldsymbol{F}_{gg}, \boldsymbol{F}_{go}, \boldsymbol{F}_{oo}\}$. We also vary the number of global prototypes $m_g$ and focused prototypes $m_f$ to see the influence of feature encoding on each group of prototypes.

Results can be seen in Tab. 3. From the experiments with $m_g = m_f = 8$, the SSv2$^\sharp$ dataset gets much improvement from the use of object-object feature $\boldsymbol{F}_{oo}$, while the Kinetics dataset benefits more from the global-global feature $\boldsymbol{F}_{gg}$. This is reasonable since SSv2 dataset includes more actions with multiple objects. From the experiments with $m_g = 16$, the global prototypes seem to work equally well using the three encoded features on both datasets. The experiments with $m_f = 16$ suggest that focused prototypes work better with global-global relations. When using all three features (last row of each block), the comparison between different choices of $m_g$ and $m_f$ indicates that the two groups of prototypes capture complementary aspects of the action, since the performances got significantly improved when two groups of prototypes both present. A more detailed figure showing the accuracy difference of each action class can be found in the supplementary material.

### 4.3   Analysis of compound prototypes

The core of our proposed method is the generation and matching of compound prototypes. In this section we conduct extensive experiments to get a more comprehensive understanding of the two groups of prototypes.

**What aspect of the action does each prototype capture?** We address this question by investigating the self-attention operation that generates the prototypes. From Equation 5, the self-attention weight $\boldsymbol{\alpha} \in \mathbb{R}^{(B+2)\times T}$ on each frame represents from which part of the video does each prototype gather its information. To better understand the prototypes, we visualize this attention in Fig. 2 using two 1-shot 2-way examples. In the visualization we average the $B + 2$ attention weights in each frame, forming $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^T$, and show this averaged value on each of the $T = 8$ frames. For clarity we only show 2 global prototypes and 2 focused prototypes in each video. We also show the video similarity scores and the similarity of matched prototypes at the bottom of the figure.

The visualization is shown in Fig. 2. We first analyze the attention of each prototype. In all videos, the global prototype $\boldsymbol{p}_{g,2}$ have high attention weights on the start of the *action* (not the start of the *video*), and $\boldsymbol{p}_{g,6}$ pays more attention to the frames that contain appearance change compared with other frames (no hand existence). This is expected since $L_{div}$ forces each global prototype to be different, while the 1-to-1 matching encourages each global prototype to focus on similar aspects so that correct video similarity can be predicted. For the focused prototypes, $\boldsymbol{p}_{f,1}$ usually gives high attention to the last few frames, and $\boldsymbol{p}_{f,2}$ pays more attention on the middle frames. This is also expected since $L_{att}$ refrains the focused prototypes to attend on similar temporal locations, and
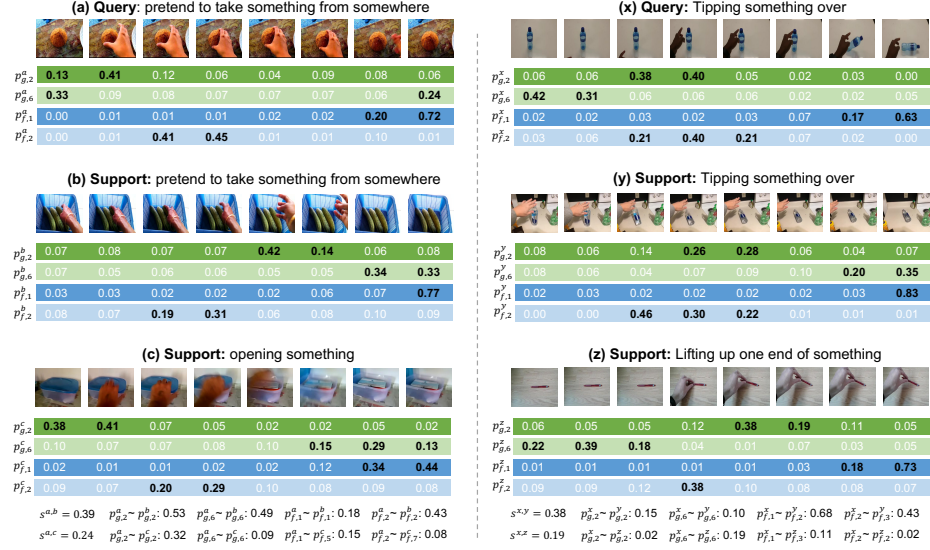
**Fig. 2.** Visualization of the self-attention weight of two global prototypes and two focused prototypes on each timestamp of the input. Attention weights higher than average (0.125) are marked in black. We can see the global prototypes capture a certain aspect of the action in the video, regardless of temporal location: $\boldsymbol{p}_{g,2}$ - the start of the action; $\boldsymbol{p}_{g,6}$ - the frames without hand. Meanwhile, the focused prototypes mainly attend on fixed timestamps of the video: $\boldsymbol{p}_{f,1}$ - the end of the video; $\boldsymbol{p}_{f,2}$ - the middle part of the video. Example to the left comes from the SSv2° dataset and the example to the right is from SSv2$^\sharp$. Video similarity scores $s$ and similarity scores of matched prototypes $\boldsymbol{p}_* \sim \boldsymbol{p}_*$ are shown at the bottom.

bipartite matching allows similar actions to be matched even when they are at different temporal locations of the videos. A similar phenomenon exists in the object detection task [6], where each object query focuses on detecting objects in a specific spatial location of the image.

In the left example, both the prototype pairs $< \boldsymbol{p}_{g,2}^a, \boldsymbol{p}_{g,2}^b >$ and $< \boldsymbol{p}_{g,6}^a, \boldsymbol{p}_{g,6}^b >$ have high similarity scores (0.53 and 0.49 shown at the bottom of the left example). This indicates that video $a$ and $b$ have similar starts, and the intra-video appearance change is also similar. Thus the query action is correctly classified as "Pretending to take something from somewhere". In the right example, we can see the effectiveness of the focused prototypes. By Hungarian matching, $\boldsymbol{p}_{f,1}^x$ is matched with $\boldsymbol{p}_{f,2}^y$. Since they both encode the frames where the hands just tip the objects over, these two prototypes give high similarities, enabling the query action of "Tipping something over" to be correctly recognized.

A statistical analysis of self-attention weights can be found in Fig. 3 showing the average response of the first 4 global prototypes and the first 4 focused prototypes on all videos of the test set. As a result of the loss functions $L_{div}, L_{att}$ and the matching strategies, on both SSv2$^\sharp$ and Kinetics datasets, $\boldsymbol{P}_g$ (first 4 rows)
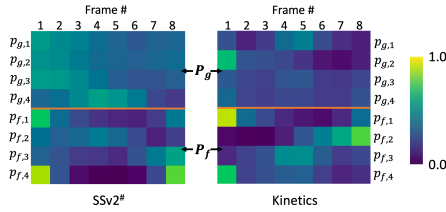
**Fig. 3.** Visualization of self-attention weights of $\boldsymbol{P}_s$ and $\boldsymbol{P}_d$ for all samples on the test set of SSv2$^\sharp$ and Kinetics datasets.
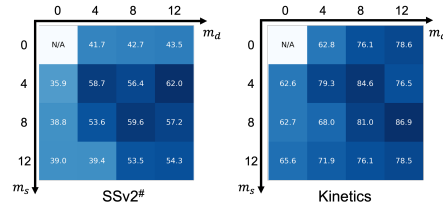
**Fig. 4.** Performance on the SSv2$^\sharp$ and Kinetics datasets when changing the number of global/focused prototypes.

have a more uniform attention distribution, while $\boldsymbol{P}_f$ have obvious temporal regions of focus. The diversity of the prototypes ensures a robust representation of the videos, thus similarity between videos can be better computed during the few-shot learning process.

**How much does each group of prototype contribute?** To find the answer, we test our method using different numbers of prototypes ($m_g$ and $m_f$) and show the results in Fig. 4. Although the best combination of $m_g$ and $m_f$ are different for each dataset, the performance gets better when the number of prototypes increases, and after a certain threshold, the result saturates because of the overfitting on the training data. Best results on both SSv2$^\sharp$ (62.0) and Kinetics (86.9) datasets are achieved when $m_f$ is larger than $m_g$. Although $m_g = m_f = 8$ is not the optimal setting, we apply this setting in Sec. 4.1 and Sec. 4.2 since it is the most stable setting on all datasets. A method to automatically choose the number of prototypes is left for our future work.

Also, we show the class accuracy improvement when our method uses both groups of prototypes compared with our method using only one group of prototype. In Fig. 5, orange bars denote the accuracy difference between the $m_g = m_f = 8$ setting and the $m_g = 16, m_f = 0$ setting, which indicates the performance gain by introducing the focused prototypes. Blue bars, on the other hand, show the accuracy improvement brought by the global prototypes. We can see on the SSv2$^\sharp$ dataset that when combining the two groups of prototypes, some hard classes like "pulling S out of S", "pulling S from left to right" and "pushing S from right to left" can be better distinguished. From the results of the two datasets, we observe the focused prototypes are more effective in the Kinetics dataset. This is because the Kinetics dataset focuses more on appearance, which can be better captured and compared by the focused prototypes.

**Will wrong bipartite matching destroy the temporal ordering of actions?** Although we observe great performance gain brought by $\boldsymbol{P}_f$ in Tab. 3 and Fig. 5, the bipartite matching will unavoidably produce some wrongly matched prototype pairs when creating the correct matchings. Our additional experi-
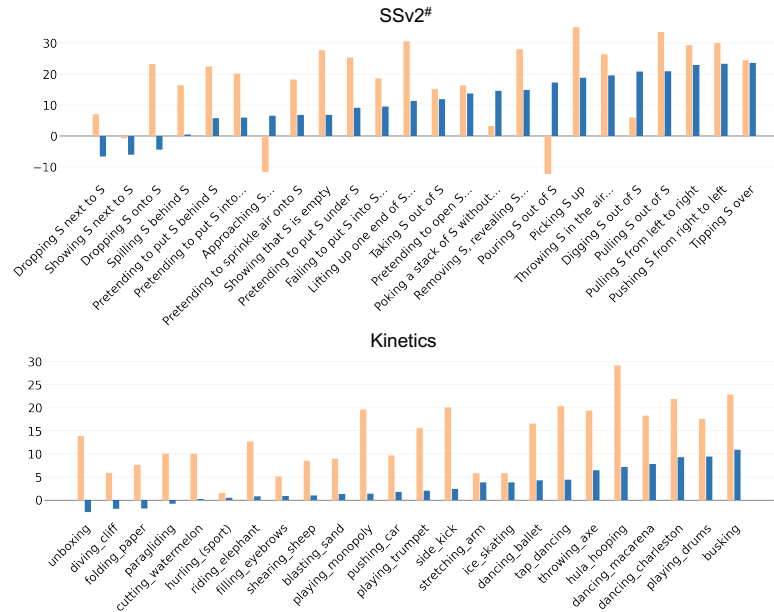
**Fig. 5.** Class accuracy improvement when our method uses $m_g = m_f = 8$ prototypes compared to: orange bars: $m_g = 16$, $m_f = 0$; blue bars: $m_g = 0$, $m_f = 16$. S stands for the abbreviation of "something".

ments in the supplementary show that filtering the matched prototypes with low similarity negatively affects the convergence of the model. One reason is that positional encoding implicitly encodes the temporal ordering of the frames within each prototype. During training, the similarity scores of all the wrong matching pairs are learned to be small and so that the final similarity score can be dominated by the similarity score of the correctly matched prototype pairs.

## 5    Conclusion

In this work, we introduce a novel method for few-shot action recognition by generating global and focused prototypes and compare video similarity based on the prototypes. When generating the prototypes, we encode spatiotemporal object relations to address the actions that involve multiple objects. The two groups of prototypes are encouraged to capture different aspects of the input by different loss functions and matching strategies. In our future work, we will explore a more flexible prototype matching strategy that can avoid the mismatch in the bipartite matching.

# References

1. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. In: NeurIPS (2016) 3
2. Antoniou, A., Edwards, H., Storkey, A.: How to train your maml. ICML (2019) 3
3. Bishay, M., Zoumpourlis, G., Patras, I.: Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. BMVC (2019) 3
4. Cao, C., Li, Y., Lv, Q., Wang, P., Zhang, Y.: Few-shot action recognition with implicit temporal alignment and pair similarity optimization. CVIU (2021) 8
5. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: CVPR (2020) 2, 3, 4, 7, 8, 9
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020) 4, 6, 12
7. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017) 3, 7
8. Chang, C.Y., Huang, D.A., Sui, Y., Fei-Fei, L., Niebles, J.C.: D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In: CVPR (2019) 3
9. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: CVPR (2018) 2
10. Chowdhury, A., Jiang, M., Chaudhuri, S., Jermaine, C.: Few-shot image classification: Just use a library of pre-trained feature extractors and a simple classifier. In: ICCV (2021) 1
11. Cong, Y., Liao, W., Ackermann, H., Rosenhahn, B., Yang, M.Y.: Spatial-temporal transformer for dynamic scene graph generation. In: ICCV (2021) 6
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 8
13. Deng, J., Yang, Z., Chen, T., Zhou, W., Li, H.: Transvg: End-to-end visual grounding with transformers. ICCV (2021) 4
14. Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. ICLR (2019) 3
15. Doersch, C., Gupta, A., Zisserman, A.: Crosstransformers: spatially-aware few-shot transfer. NeurIPS (2020) 4
16. Fan, Q., Zhuo, W., Tang, C.K., Tai, Y.W.: Few-shot object detection with attention-rpn and multi-relation detector. In: CVPR (2020) 1
17. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning. PMLR (2017) 3, 4
18. Fu, Y., Zhang, L., Wang, J., Fu, Y., Jiang, Y.G.: Depth guided adaptive meta-fusion network for few-shot video recognition. In: ACM MM (2020) 1, 3, 9
19. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., et al.: The" something something" video database for learning and evaluating visual common sense. In: ICCV (2017) 3, 7
20. Grauman, K., Westbury, A., Byrne, E., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. arXiv preprint arXiv:2110.07058 (2021) 1
21. Gui, L.Y., Wang, Y.X., Ramanan, D., Moura, J.M.: Few-shot human motion prediction via meta-learning. In: ECCV (2018) 3

22. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) 2, 3, 4, 5, 8
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 8
24. Hong, J., Fisher, M., Gharbi, M., Fatahalian, K.: Video pose distillation for few-shot, fine-grained sports action recognition. In: ICCV (2021) 3
25. Huang, Y., Cai, M., Li, Z., Sato, Y.: Predicting gaze in egocentric video by learning task-dependent attention transition. In: ECCV (2018) 1
26. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: ICCV (2019) 1
27. Kang, D., Kwon, H., Min, J., Cho, M.: Relational embedding for few-shot classification. In: ICCV (2021) 3
28. Kliper-Gross, O., Hassner, T., Wolf, L.: One shot similarity metric learning for action recognition. In: SIMBAD (2011) 1
29. Koch, G., Zemel, R., Salakhutdinov, R., et al.: Siamese neural networks for one-shot image recognition. In: ICML (2015) 3
30. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV (2011) 3, 7
31. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly (1955) 7
32. Kumar Dwivedi, S., Gupta, V., Mitra, R., Ahmed, S., Jain, A.: Protogan: Towards few shot learning for action recognition. In: CVPRW (2019) 3
33. Li, H., Eigen, D., Dodge, S., Zeiler, M., Wang, X.: Finding task-relevant features for few-shot learning by category traversal. In: CVPR (2019) 1
34. Li, S., Liu, H., Qian, R., Li, Y., See, J., Fei, M., Yu, X., Lin, W.: Ta2n: Two-stage action alignment network for few-shot action recognition. arXiv preprint arXiv:2107.04782 (2021) 2, 3
35. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 8
36. Liu, W., Zhang, C., Lin, G., Liu, F.: Crnet: Cross-reference networks for few-shot segmentation. In: CVPR (2020) 1
37. Liu, Y., Zhang, X., Zhang, S., He, X.: Part-aware prototype network for few-shot semantic segmentation. In: ECCV (2020) 1
38. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. ICCV (2021) 4, 6
39. Lu, Z., He, S., Zhu, X., Zhang, L., Song, Y.Z., Xiang, T.: Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In: ICCV (2021) 6
40. Luo, Z., Guillory, D., Shi, B., Ke, W., Wan, F., Darrell, T., Xu, H.: Weakly-supervised action localization with expectation-maximization multi-instance learning. In: ECCV (2020) 2
41. Ma, J., Gorti, S.K., Volkovs, M., Yu, G.: Weakly supervised action selection learning in video. In: CVPR (2021) 2
42. Mishra, A., Verma, V.K., Reddy, M.S.K., Arulkumar, S., Rai, P., Mittal, A.: A generative approach to zero-shot and few-shot action recognition. In: WACV (2018) 3
43. Patravali, J., Mittal, G., Yu, Y., Li, F., Chen, M.: Unsupervised few-shot action recognition via action-appearance aligned meta-adaptation. In: ICCV (2021) 3

44. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal-relational crosstransformers for few-shot action recognition. In: CVPR (2021) 2, 3, 4, 8, 9
45. Qiao, S., Liu, C., Shen, W., Yuille, A.L.: Few-shot image recognition by predicting parameters from activations. In: CVPR (2018) 3
46. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. ICLR (2017) 3
47. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. NeurIPS (2017) 1, 3
48. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012) 7
49. Sun, R., Li, Y., Zhang, T., Mao, Z., Wu, F., Zhang, Y.: Lesion-aware transformers for diabetic retinopathy grading. In: CVPR (2021) 6
50. Thatipelli, A., Narayan, S., Khan, S., Anwer, R.M., Khan, F.S., Ghanem, B.: Spatio-temporal relation modeling for few-shot action recognition. arXiv preprint arXiv:2112.05132 (2021) 3
51. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 2, 4, 5, 6
52. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. NeurIPS (2016) 3, 4, 8, 9
53. Wang, H., Zhang, X., Hu, Y., Yang, Y., Cao, X., Zhen, X.: Few-shot semantic segmentation with democratic attention networks. In: ECCV (2020) 1
54. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: Panet: Few-shot image semantic segmentation with prototype alignment. In: ICCV (2019) 1
55. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV (2016) 5
56. Wang, X., Ye, W., Qi, Z., Zhao, X., Wang, G., Shan, Y., Wang, H.: Semantic-guided relation propagation network for few-shot action recognition. In: ACM MM (2021) 3, 8
57. Wang, X., Huang, T.E., Darrell, T., Gonzalez, J.E., Yu, F.: Frustratingly simple few-shot object detection. ICML (2020) 3
58. Wei, X.S., Wang, P., Liu, L., Shen, C., Wu, J.: Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. TIP (2019) 3
59. Xian, Y., Korbar, B., Douze, M., Schiele, B., Akata, Z., Torresani, L.: Generalized many-way few-shot video classification. In: ECCV (2020) 3
60. Xian, Y., Korbar, B., Douze, M., Torresani, L., Schiele, B., Akata, Z.: Generalized few-shot video classification with video retrieval and feature generation. TPAMI (2021) 3
61. Xu, B., Ye, H., Zheng, Y., Wang, H., Luwang, T., Jiang, Y.G.: Dense dilated network for few shot action recognition. In: ICMR (2018) 3
62. Xu, C., Fu, Y., Liu, C., Wang, C., Li, J., Huang, F., Zhang, L., Xue, X.: Learning dynamic alignment via meta-filter for few-shot learning. In: CVPR (2021) 1
63. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: CVPR (2020) 6
64. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers. NeurIPS (2021) 4
65. Ye, H.J., Hu, H., Zhan, D.C., Sha, F.: Few-shot learning via embedding adaptation with set-to-set functions. In: CVPR. pp. 8808–8817 (2020) 4, 8, 9

66. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: ICCV (2019) 6
67. Zhang, C., Cai, Y., Lin, G., Shen, C.: Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In: CVPR (2020) 3
68. Zhang, C., Gupta, A., Zisserman, A.: Temporal query networks for fine-grained video understanding. In: CVPR (2021) 4
69. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P.H., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: ECCV (2020) 3, 7, 8, 9
70. Zhang, S., Zhou, J., He, X.: Learning implicit temporal alignment for few-shot video classification. IJCAI (2021) 2, 3, 8, 9
71. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: ECCV (2018) 8, 9
72. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: ECCV (2018) 3, 7
73. Zhu, L., Yang, Y.: Label independent memory for semi-supervised few-shot video classification. TPAMI (2020) 1, 3, 8, 9
74. Zhu, X., Toisoul, A., Perez-Rua, J.M., Zhang, L., Martinez, B., Xiang, T.: Few-shot action recognition with prototype-centered attentive learning. BMVC (2021) 1, 3, 8
75. Zhu, Z., Wang, L., Guo, S., Wu, G.: A closer look at few-shot video classification: A new baseline and benchmark. BMVC (2021) 3