Few-shot Action Recognition with Prototype-centered Attentive Learning

Xiatian Zhu
xiatian.zhu@samsung.com
Antoine Toisoul
atoisoul@fb.com
Juan-Manuel Pérez-Rúa
jmpr@fb.com
Brais Martinez
brais.a@samsung.com
Li Zhang
lizhangfd@fudan.edu.cn
Tao Xiang
t.xiang@surrey.ac.uk

Samsung AI Centre, Cambridge

Abstract

Few-shot action recognition aims to recognize new action classes with few training samples. Most existing methods adopt a meta-learning approach with episodic training. In each episode, the few samples in a meta-training task are split into support and query sets. The former is used to build a classifier, which is then evaluated on the latter using a query-centered loss for model updating. There are however two major limitations: lack of data efficiency due to the query-centered only loss design and inability to deal with the support set outlying samples and inter-class distribution overlapping problems. In this paper, we overcome both limitations by proposing a new Prototype-centered Attentive Learning (PAL) model composed of two novel components. First, a prototype-centered contrastive learning loss is introduced to complement the conventional query-centered learning objective, in order to make full use of the limited training samples in each episode. Second, PAL further integrates a hybrid attentive learning mechanism that can minimize the negative impacts of outliers and promote class separation. Extensive experiments on four standard few-shot action benchmarks show that our method clearly outperforms previous state-of-the-art methods, with the improvement particularly significant on the most challenging fine-grained action recognition benchmark.

1 Introduction

The universal adoption of smartphones has led to a large quantity of videos being produced and shared on social media platforms, which are in urgent need of automated analysis. As a result, video action recognition [4, 16, 28, 33] has been studied intensively with a recent focus on fine-grained actions [11]. Most recent action recognition models employ deep

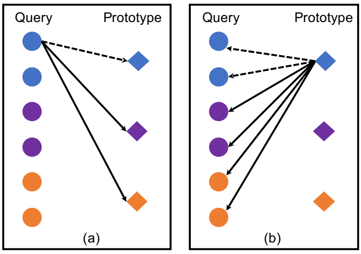


Figure 1: Illustration of (a) conventional query-centered learning vs. (b) our prototype-centered learning in a 3-way 2-shot case. The former classifies a query sample (cycle) against the space of prototype (diamond) via computing pairwise similarity. In contrast, the latter considers an opposite direction by comparing a prototype with all the query samples, pulling close those of the same class whilst pushing way the others. Each class is color coded. Dashed and solid arrows represent pairs from same and different classes.

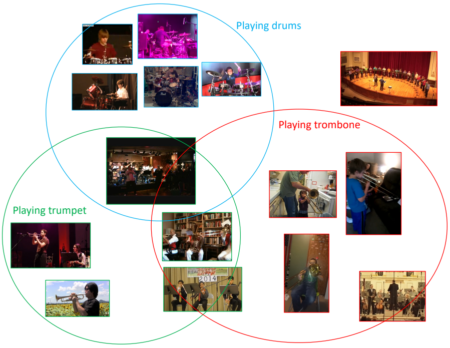


Figure 2: Illustration of the outlying video samples and inter-class overlapping problems in a support set containing three music-instrument-playing classes. There is a clear outlier in the trombone class due to its unusual view point. There is also plenty of class overlapping due to the intrinsic visual similarity between the three classes, particularly when these instruments are played in groups.

convolutional neural networks (CNNs) which are known to be data hungry as they require a large number (at least hundreds) of annotated samples for each action class to be trained. However, collecting and annotating such a large amount of data is expensive, tedious and sometimes even infeasible for some rare fine-grained action classes. This is the reason why few-shot action recognition has started to receive increasing interest [2, 37, 40] as it aims at constructing a video action classifier with few training samples (*e.g.*, 1-5) per class.

Few-shot action recognition is a special case of few-shot learning (FSL). Most FSL methods follow a meta-learning (or learning-to-learn) paradigm [24, 30] characterized by episodic training which aims to learn a model or optimizer from a set of base/seen tasks, in order to generalize well to new tasks with few labeled training samples/shots. Specifically, a meta-training set containing abundant training samples per seen/base class is used to sample a large number of training episodes. In each episode, the training data is split into a support set with N classes and K samples per class to mimic the setting of target meta-test tasks, and a query set from the same N classes. A classifier is built using the support set and then evaluated on each query set sample with a classification loss for model updating. Many state-of-the-art FSL methods [34, 36] are based on the popular prototypical network (ProtoNet) [24] for its simplicity and effectiveness. With ProtoNet, the mean of each support set class is computed as a class prototype for classifying each query sample (see Figure 1(a)).

There are however two major limitations in existing few-shot action recognition methods [2, 37, 40]. The first limitation is the lack of data efficiency due to the query-centered learning objective adopted by existing methods. This is shown in Figure 1(a): with only K samples for each of the N classes in the support set, these $N \times K$ samples are first reduced to N prototypes; a loss term (*e.g.*, cross-entropy) is then computed for each query sample individually based on its distances to the prototypes, without any consideration on how the whole query set samples should be distributed. This query-centered learning objective thus does not make full use of the limited training data in each episode.

The second limitation is their inability to address two fundamental challenges of FSL: outlying samples and inter-class distribution overlapping in the support set. These issues are in general more severe in video data than imagery due to extra spatio-temporal dynamics. As illustrated in Figure 2, outliers are caused by unusual viewpoint, background, occlusion etc. These are problematic for any learning task, but particularly for few-shot action recognition as with few samples per class, a single outlier can have an immense impact on the class distribution. The problem of inter-class overlapping is also common when the training samples of different classes have very similar background, or just being visually similar. This problem is especially acute for fine-grained action classes. Both problems are not addressed when a class is simply represented as class mean without considering both the intra-class relationships to identify outliers and inter-class relationships to avoid class overlapping.

To overcome the aforementioned limitations, we propose a novel *Prototype-centered Attentive Learning* (PAL) model with two key components. First, a *prototype-centered contrastive learning* loss is introduced. This loss is computed by comparing a given prototype against all the query-set samples for discriminative learning (see Figure 1(b)). This new learning objective is complementary to conventional query-centered learning objective, and combining the two enables PAL to make a full use of limited training data and hence improve data efficiency during training. Second, a *hybrid attentive learning* component is developed consisting of self-attention on support-set samples and cross-attention from query-set to support-set samples. As an ***inductive*** design, it aims to mitigate the negative effect of outlying samples and promote class separation. Importantly, it can be seamlessly integrated with the query- and prototype-centered learning objectives in a unified meta-learning pipeline.

In this paper, we make the following **contributions**: (1) We propose a novel Prototype-centered Attentive Learning (PAL) for few-shot action recognition, specifically designed to address the data efficiency, outliers and class overlapping problems that existing methods suffer from. (2) We introduce a novel prototype-centered contrastive learning objective to complement the existing query-centered objective, in order to make full use of the limited training data available in few shot learning. (3) We further introduce a hybrid attentive learning strategy which is dedicated for solving the under-studied outlier sample and class overlapping problems. (4) Extensive experiments show that our model achieves new state-of-the-art results on four few-shot action datasets. Its performance is particularly compelling on the more challenging fine-grained benchmark, yielding around 10% improvement.

2 Related work

Action recognition Previous efforts have been focused on training action recognition models on large video datasets (*e.g.*, the coarse-grained Kinetics [4] and fine-grained Something-Something [11]). Computationally, utilizing 2D CNNs [16, 23, 31] for action recognition is more efficient than using their 3D counterparts [4, 16, 28, 33]. In particular, Temporal Segment Networks (TSN) [31] are one of the most popular action classification models. They work by extracting features from temporally sparsely sampled frames with a 2D CNN before applying an average pooling to obtain a video-level feature representation and a prediction. We use TSN as the feature extractor in our few-shot action recognition model as we found that more complex feature extractors [4, 16, 28, 33] do not fare better performance but are computationally more demanding.

Few-shot action recognition When the research focus in action recognition shifted towards fine-grained action classes, the problem of collecting sufficient training samples per class started to become an obstacle. Few-shot action recognition emerged as a potential solution to this problem [2, 37, 40]. CMN [40] employs a compound memory network to store the representation and classify videos by matching and ranking. OTAM [2] measures the query-centered distance with respect to support set samples, by explicitly leveraging the temporal ordering information in the query video via ordered temporal alignment. ARN [37] learns a query-centered similarity between the query and support video clips with a pipeline following RelationNet [26]. It exploits augmentation-guided spatial and temporal attention with auxiliary self-supervision training losses. Concurrently introduced along with our work, TRX [20] leverages sub-sequence level self-attention to form query specific prototypes at varying temporal scales. This allows to mitigate the temporal misalignment problem in a partial matching manner. Despite their differences in model design, all deploy a meta-learning framework with a query-centered learning objective, thus being unable to make full use of the limited training data available in each episode. Further, none is designed to address the the outlying sample and class overlapping problems, as our model does.

Few-shot learning Few-shot action recognition is a special case of few-shot learning (FSL). Most existing FSL models focus on recognizing static images. They can be roughly categorized into two groups including metric-based methods [7, 15, 18, 19, 24, 26, 30, 34, 36] and optimization-based methods [9, 21]. All existing few-shot action recognition methods and our PAL belong to the first group, where what is meta-learned is a feature embedding network for videos. Note that the idea of using self-attention for few-shot learning has previously been explored in [34]. However, the model in [34] only applies self-attention to class prototypes whereas, in contrast, our model applies it to the support set as well as query samples. This gives our model the capability of dealing with sample outliers while improving data efficiency. Following the practice of most recent FSL methods [34, 36], we also incorporate pretraining of the feature embedding network into our model.

Self-attention Our attentive learning is based on self-attention across data instances, which has been first introduced in transformers as a global self-attention module for machine translation tasks [29]. Non-local neural networks [32] applied the core self-attention block from transformers for context modeling and feature learning in computer vision tasks. They learn an affinity map among all pixels in the image that allows the neural network to effectively increase the receptive field to the global context. State-of-the-art performances have been shown in classification [8], self-supervised learning [5], semantic segmentation [10, 38], object detection [3, 35, 41] by using such a transformer-based attention model. Different from these works, in this paper, we propose a hybrid attentive learning mechanism that aims to exploit the support set *self-attention* and query-to-support *cross-attention* in a unified meta-learning manner.

3 Method

Problem definition We consider the few-shot video action classification problem [2, 40]. Given a meta-test dataset D_{test}, we sample a N-way K-shot classification task to test a learned FSL action model $\boldsymbol{\theta}$. To train the model $\boldsymbol{\theta}$ in a way that it can perform well on those sampled classification tasks, episodic training is adopted to meta-learn the model. Concretely, a large

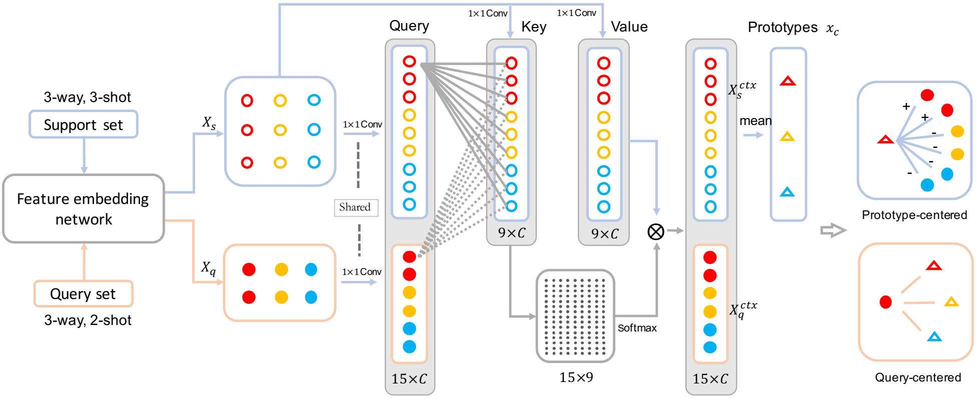


Figure 3: Overview of our Prototype-centered Attentive Learning (PAL). Given a training episode including a support set (3-way 3-shot) and a query set (2 samples per class), a CNN is first used to extract their feature vectors $\mathbf{X}_s$ and $\mathbf{X}_q$. $\mathbf{X}_s$ is then used to form the *Key* and *Value*. For *support-set self-attention* learning, $\mathbf{X}_s$ is set as the *Query*. For *query-to-support cross-attention learning*, $\mathbf{X}_q$ is instead set as *Query*. The two processes output attentive support-set $\mathbf{X}_s^{\text{ctx}}$ and query-set $\mathbf{X}_q^{\text{ctx}}$ features. Forming per-class prototypes $\mathbf{x}_c$ from $\mathbf{X}_s^{\text{ctx}}$ as feature mean, $\mathbf{X}_q^{\text{ctx}}$ can be classified (*i.e.*, query-centered learning). Further, we conduct a novel prototype-centered contrastive learning in an opposite direction by comparing a given prototype with all the query-set samples and conducting discriminative contrastive learning.

number of N-way K-shot tasks are randomly sampled from a meta-training set D_{train}, and then used to train the model in an episodic manner. In each episode, we start by sampling N classes from D_{train} at random. Then we randomly draw labeled training samples from these classes to create a *support* set $\mathcal{S}$ and a *query* set $\mathcal{Q}$ consisting of K and Q samples per class, respectively. Formally, the support and query sets are defined as:

$$\mathcal{S} = \{(x_i, y_i) \mid y_i \in C\}_{i=1}^{NK}, \quad \mathcal{Q} = \{(x_i, y_i) \mid y_i \in C\}_{i=1}^{NQ}. \tag{1}$$

Note, $\mathcal{S} \cap \mathcal{Q} = \phi$ are sample-wise non-overlapping.

Typically, episodic training is conducted in a two-loop manner [24]: the support set is used in the inner loop to construct a classifier for the N classes, and the query set is then used in the outer loop to evaluate this classifier and update the model parameters $\boldsymbol{\theta}$. It is noteworthy that as the objective is to obtain a learner able to recognize novel classes, D_{train} and D_{test} are set to be disjoint in the class space, *i.e.*, $D_{\text{train}} \cap D_{\text{test}} = \phi$.

3.1 Pretraining

As in all existing few-shot action classification models [2, 37, 40], one of the key objectives is to learn a feature embedding network through meta learning, so that it can generalize to any unseen action class. The embedding networks adopted are based on those used by existing video action recognition model with TSN [31] being the most popular choice. For example, in the state-of-the-art OTAM [2] model, a ResNet-50 based TSN model, pre-trained on ImageNet, was meta-trained together with a time-warping based video distance. However, in most recent FSL methods for static image classification [34, 36], pretraining the embedding network on the whole meta-training set before episodic training starts has become a musthave step. In this work, we also adopt such a pretraining step and show in our experiments that this step is vital (see Sec. 4).

Specifically, we use as our feature embedding network a TSN action model [31]. It is then pretrained on the whole training set D_{train} with a cosine similarity based cross-entropy loss. Given a video sample $x_i \in D_{\text{train}}$ with a varying length and redundant temporal information, we first sample video frames. We adopt the same sparse sampling strategy as introduced in [31]: splitting the video into T equal-sized segments and randomly selecting one frame from each segment. This way, each video can be represented using a fixed number of frames $x_i = \{f_1, f_2, \cdots, f_T\}$ where f_j is a random frame from the j-th segment. As sampled frames cover the majority of original time span, long-term temporal information is kept whereas the spatio-temporal redundant information is reduced significantly.

Next, each sampled video frame f_j is individually encoded by a feature embedding network $h()$ and classified by a cosine-distance based classifier, giving us frame-level classification score vectors $\mathbf{s}_j^f = [s_1^f, s_2^f, \cdots, s_Z^f]$ with Z the total number of classes in D_{train}. In particular, for each frame, $s_z^f = \cos(h(f_j),\ \mathbf{w}_z)$ where $\cos()$ denotes the cosine similarity function and $\mathbf{w}_z$ ($z \in \{1, \cdots, Z\}$) the classifier weights of each class. Then, these per-frame score vectors are averaged to get video-level scores. Finally, a softmax is applied to video-level scores to get video-level action probabilities.

After the pretraining stage, the feature embedding network together with a cosine distance can be used directly for meta-test without going through the episodic training stage. Our experiments show that this turns out to be a surprisingly strong baseline that even achieves better results than current state-of-the-art OTAM [2] (see Table 1). This verifies for the first time that a good feature embedding (or feature reuse) is also critical for few-shot action modeling – a finding that has been reported in recent static image FSL works [6, 17, 27]. Nonetheless, this baseline is still limited for FSL since it lacks a "learning to learn" or task adaptation capability to better deal with unseen new tasks. To this end, we next introduce our *Prototype-centered Attentive Learning* (PAL) method. The overview of PAL is depicted in Figure 3.

3.2 Hybrid attentive learning

Hybrid attentive learning (HAL) is designed to mitigate the inter-class ambiguity and the intra-class outlying sample problem by allowing each support set sample to examine all other samples in order to identify and fix both problems. It relies on a transformer self-attention module whose parameters are meta-learned together with those of the embedding network $h()$. Once learned, during meta-test, it is used to exploit task-specific contextual information for superior task adaptation. Concretely, given an episode we first extract a d-dimensional feature representation using a TSN $h()$. The video-level feature vectors are obtained for both support-set and query-set by average pooling along frames.

Support set self-attention Formally, let $\mathbf{X}_s \in R^{NK \times d}$ and $\mathbf{X}_q \in R^{NQ \times d}$ be the support-set and query-set feature matrix respectively. The input to HAL is in the triplet form of (`Query`, `Key`, `Value`). To learn discriminative contextual information per episode, the input is designed based on the support-set samples as:

$$\text{Query} = \mathbf{X}_s \mathbf{W}_Q, \ \ \text{Key} = \mathbf{X}_s \mathbf{W}_K, \ \ \text{Value} = \mathbf{X}_s \mathbf{W}_V, \tag{2}$$

where $\mathbf{W}_Q/\mathbf{W}_K/\mathbf{W}_V \in R^{d \times d_a}$ are learnable parameters (each represented by a fully connected layer) that project the TSN feature to a d_a-D latent space. As `Query`, `Key` and `Value` share the same input source (*i.e.*, support-set data), *self-attention* can be formulated as:

$$\mathbf{X}_s^{\text{ctx}} = \mathbf{X}_s + \text{softmax}(\frac{\mathbf{X}_s\mathbf{W}_Q(\mathbf{X}_s\mathbf{W}_K)^\top}{\sqrt{d_a}})(\mathbf{X}_s\mathbf{W}_V), \tag{3}$$

where softmax() is a row-wise softmax function for attention normalization. Residual learning is adopted for a more stable model convergence. As written in Eq (3), pairwise similarity defines the attentive scores between any two support-set samples, and further used for weighted aggregation in the `Value` space. The intuition is that, statistically sample pairs from the same class often enjoy more similarity than those with different classes except for very few outlier instances. As a result, this attentive learning reinforces the importance of class-sensitive information, subject to the context of current task's classes. Consequently, the effect caused by class irrelevant information of outlier samples will be well controlled during feature transformation. Finally, intra-class variation shrinks and inter-class ambiguity can be reduced accordingly.

Query set cross-attention For consistency, the query-set samples should be also contextualized in a task-specific manner, because they will be classified into a label space formed by support-set samples. To that end, we introduce a *cross-attention* process from query-set samples $\mathbf{X}_q$ to support-set samples $\mathbf{X}_s$ formulated as:

$$\mathbf{X}_q^{\text{cxt}} = \mathbf{X}_q + \text{softmax}(\frac{\mathbf{X}_q\mathbf{W}_Q(\mathbf{X}_s\mathbf{W}_K)^\top}{\sqrt{d_a}})(\mathbf{X}_s\mathbf{W}_V). \tag{4}$$

By doing this, in the same spirit as support-set self-attention, query-set samples are also enriched by contextual information. Note that each query set sample transformation is done independently from other query samples as the `Keys` and `Values` are provided by the support set only. Our model thus remains ***inductive*** as existing FSL action models.

Learning objective To obtain the supervision for our HAL module, we adopt the popular prototype based objective loss [24]. With the attentive support-set feature matrix $\mathbf{X}_s^{\text{ctx}}$, we first form the prototype for each class c as the mean feature: $\mathbf{w}_c = \frac{1}{K}\sum_{i=1}^{NK} I(y_i == c)\mathbf{X}_s^{\text{ctx}}(i,:)$, where y_i denotes the class label of i-th support-set sample. We then compute the cosine similarity between any query-set feature $\mathbf{X}_q^{\text{ctx}}(j,:)$ and all prototypes $\{\mathbf{w}_c\}_{c=1}^N$ and obtain the classification probability vector $\tilde{\mathbf{p}}_j = [\tilde{p}_1, \cdots, \tilde{p}_N] \in R^N$ by softmax function. With the query-set class labels y_j, a cross-entropy loss can be derived as the meta-training objective.

3.3 Prototype-centered contrastive learning

To further improve discriminative learning, we further introduce a complementary prototype-centered contrastive learning. More specifically, for a prototype $\mathbf{w}_c$, we define the query-set samples $\mathcal{Q}_c$ from the same class as the positive matches and all the others as the negative. We then compute the cosine similarity between $\mathbf{w}_c$ and all the query-set samples, and devise the following prototype centered contrastive loss function:

$$\mathcal{L}_{pcc} = \frac{1}{N}\sum_{c=1}^{N} -\log\Big(\frac{\sum_{i\in\mathcal{Q}_c}\cos(\mathbf{w}_c, \mathbf{X}_q^{\text{ctx}}(i,:))}{\sum_{j\in\mathcal{Q}}\cos(\mathbf{w}_c, \mathbf{X}_q^{\text{ctx}}(j,:))}\Big), \tag{5}$$

where N is the number of class prototypes. This design attempts to pull positive query-set samples closer to their corresponding prototype (*i.e.*, the anchor of our loss), whilst pushing away the negative ones. This helps to further reduce intra-class variation while simultaneously better separate different classes.

3.4 Overall objective function

The overall objective loss function of our PAL model for meta-training is defined as: $\mathcal{L} = \mathcal{L}_{meta} + \lambda\mathcal{L}_{pcc}$, where λ is a weight hyper-parameter which we set to 1.

4 Experiments

Datasets We used four few-shot action benchmarks including both fine and coarse action classes in our evaluations. (1) *Kinetics-100* is a 100-class subset of Kinetics-400 [13], and was introduced firstly in [40] for few-shot action classification. We followed the same protocol: 64/12/24 classes and 6389/1199/2395 videos for meta-training, meta-validation and meta-testing respectively. Whilst Kinetics is one of the most commonly evaluated datasets, visual appearance and background encapsulate most class-related information rather than motion patterns [22]. With less need for temporal modeling and involving coarse-grained action classes, it presents a relatively easy action classification task. We hence further evaluated on (2) *Sth-Sth-100* created in a similar way based on the Something-Something-V2 dataset [11]. For this dataset we adopted the same protocol as [2] where 64/12/24 classes and 66939/1925/2854 videos are included for train/val/test. By considering fine-grained actions involving human object interactions with subtle differences between different classes, it presents a significantly more challenging action recognition task. To compare with very recent models [37], we also tested two more YouTube video datasets under the same setting. (3) *HMDB-51* [14] contains 51 action classes with 6,849 videos. We used 31/10/10 action classes with 4280/1194/1292 videos for train/val/test. (4) *UCF101* [25] consists of 101 action categories with 13,320 video clips. We took 70/10/21 classes with 9154/1421/2745 videos for train/val/test. Among these four datasets, only Sth-Sth-100 provides fine-grained action classes, whilst the remaining datasets present coarse actions instead. The former is considered to be more challenging due to high potentials with sample outliers and inter-class distribution overlap, the two challenges this work aims to tackle.

Evaluation metrics We used the standard 5-way 1/5-shot FSL setting [2, 40]. We randomly sampled 150,000 episodes from meta-test set and reported the mean accuracy.

Implementation details We used an ImageNet pretrained ResNet-50 [12] as the backbone network. We replaced the original fully-connected layer with a new layer that performs cosine similarity based classification. For the first training stage, we optimized the TSN feature embedding [31] (Sec. 3.1) with SGD, with a starting learning rate at 0.001 and decaying every 30 epochs by 0.1 and a total of 70 epochs. For the second stage, we conducted meta-training of both the TSN feature backbone and our PAL model end-to-end. On Sth-Sth-100, from the initial learning rate at 0.0001 we trained a total of 35 epochs with decaying epochs at 15 and 30 and each epoch consists of 200 episodes. For the other datasets, we found that training 10 epochs sufficed, with decaying points at 5, 7 and 9. For both stages, we used a cosine based classifier model. We set the dimension of feature and latent space to $d = d_a = 2024$ (Eq. (2)). During training, we resized each video frame to the size of 256×256 from which a random 224×224 region was cropped to form the input. For three coarse-grained datasets, we applied random horizontal flip during training. However, many classes in the fine-grained Sth-Sth-100 are direction-sensitive (*e.g.*, pulling something from left to right and pulling something from right to left), horizontal flip was therefore not applied. During test, we used the center crop only to get the prediction of test data and model performance.

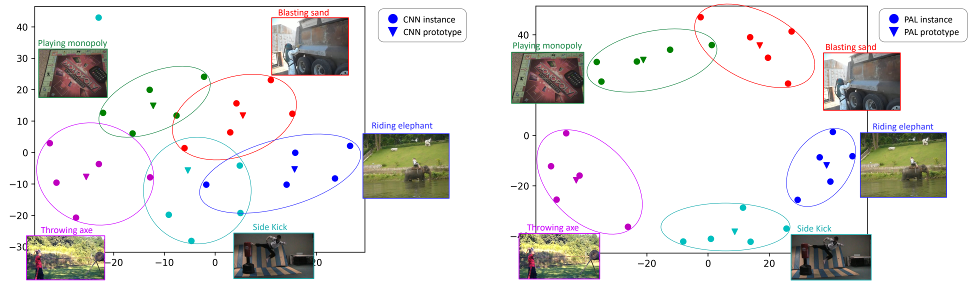
Competitors Three groups of methods are compared: (1) Classical FSL models originally

Action property	*Coarse*		*Fine*		*Coarse*		*Coarse*	
Dataset	**Kinetics-100**		**Sth-Sth-100**		**HMDB51**		**UCF101**	
	1-shot	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Matching Net† [30]	53.3	74.6	-	-	-	-	-	-
MAML† [9]	54.2	75.3	-	-	-	-	-	-
ProtoNet++* [24]	64.5	77.9	33.6	43.0	-	-	-	-
TARN* [1]	64.8	78.5	-	-	-	-	-	-
TRN++* [39]	68.4	82.0	38.6	48.9	-	-	-	-
CMN [40]	60.5	78.9	-	-	-	-	-	-
CMN++* [40]	65.4	78.8	34.4	43.8	-	-	-	-
OTAM [2]	73.0	85.8	42.8	52.3	-	-	-	-
ARN [37]	63.7	82.4	-	-	45.5	60.6	66.3	83.1
FEAT [34]	74.0	86.5	45.3	61.2	60.4	75.2	83.9	94.5
PAL (Ours)	**74.2**	**87.1**	**46.4**	**62.6**	**60.9**	**75.8**	**85.3**	**95.2**

Table 1: Few-shot action classification results. *: Results from [2]; †: Results from [40]; ProtoNet++: cosine-distance is used; TRN++: same as ProtoNet++ but with TRN feature embedding instead of TSN; CMN++: also with TRN embedding.

proposed for image classification including Matching Net [30], MAML [9] and ProtoNet [24]. All these FSL methods used the same TSN feature embedding for fair comparison. (2) Stronger action recognition models including TRN [39] and TARN [1]. (3) State-of-the-art FSL action models including CMN [40], OTAM [2], ARN [37], and FEAT [34].

Comparison to state-of-the-art The comparative results are reported in Table 1. We make the following observations: **(1)** In comparison to all classical FSL methods, the proposed PAL is clearly superior under both settings and on all datasets. **(2)** Whilst stronger action recognition models (*e.g.*, TRN++ and TARN) improve the results, they still lag behind our model by a large margin. **(3)** Interestingly, the first FSL action model CMN is shown to be inferior to both TRN++ and TARN, suggesting that its memory network's merit is less critical than better temporal structure modeling. Directly combining CMN with TRN helps little (CMN++). **(4)** By taking a pairwise temporal alignment strategy, the state-of-the-art OTAM further improves the performance. Nonetheless, it is still outperformed by our PAL particularly on the more challenging Sth-Sth-100 dataset (10.3% improvement under 5-shot). This is not surprising because OTAM's temporal warping is functionally sensitive to outlier (less consistent) support-set videos as typically encountered in fine-grained actions with human-object interactions. Besides, as compared with our model, OTAM is inferior in leveraging few shots per class during meta-learning due to the lack of prototype-centered discrimination. On the easier dataset Kinetics-100 where class overlapping is less a problem, our PAL model remains superior to OTAM by a smaller margin, indicating that tackling outlying samples is consistently a more effective strategy. **(5)** Similar to PAL, FEAT [34] also exploits a transformer [29] based self-attention for few-shot image classification. However, unlike our model, it directly adapts the class prototypes (*i.e.*, feature mean) using self-attention alone; it is thus unable to deal with outlier samples in support set. Moreover, there is no cross-attention and only conventional query-centered learning is performed during meta-training. As a result, our PAL is consistently superior to FEAT. **(6)** On the two smallest datasets HMDB51 and UCF, the proposed method also sets a new state-of-the-art, due to its superior ability to deal with sparse training samples. **(7)** Finally, we note that the best performance advantage of our PAL over all the other alternative methods is observed on Sth-Sth-100 with



((a)) Before PAL

((b)) After PAL

Figure 4: t-SNE feature distribution of support samples and per-class prototypes **(a)** before and **(b)** after our PAL in a 5-way 5-shot task on Kinetics-100. Circle indicates the variation and boundary of each class. Each class is color coded. As can be seen, the boundary of each class becomes tighter and the overlap between different classes is clearly reduced.

fine-grained human-object actions. This indicates the good effectiveness of our model in dealing with class distribution overlap and ourlier sample challenges as they are more likely involved in fine action classes.

Why PAL works? As described in the main paper, our PAL is specially designed to overcome the class overlap challenge caused by inter-class boundary ambiguity and outlier support-set samples intrinsic to new tasks. To understand the internal mechanism of our model, we visualized the change of support samples' feature representations and per-class prototypes in a new 5-way 5-shot task. Concretely, this contrasted the feature distributions with and without our PAL model to reveal how the feature space is improved. It is evident in Figure 4 that PAL does improve the separation of different classes' decision boundary by posing two effects: (1) reducing intra-class variation by mitigating the distracting effect of outlier samples in forming class decision boundary, and (2) minimizing inter-class overlap by conducting query-centered and prototype-centered discriminative learning concurrently.

5 Conclusion

In this work we have proposed a novel Prototype-centered Attentive Learning (PAL) method for few-shot action recognition. It is designed specifically to address the data efficiency and inability to deal with outliers and class-overlapping problems of existing methods. To that end, two complementary components are developed, namely prototype-centered contrastive learning that allows to make better use of few shots per class, and hybrid attention learning that aims to mitigate the negative effect of outlier support samples as well as class overlapping. They can be integrated in a single framework and trained end-to-end to maximize their compatibility. Extensive experiments validate that the proposed PAL yields new state-of-the-art performances on four action benchmarks, with the improvement on the more challenging fine-grained action recognition benchmark being the most compelling.

References

[1] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. In *BMVC*, 2019.

[2] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *CVPR*, 2020.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[5] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[7] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *NeurIPS*, 2020.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics human action video dataset. *arXiv preprint*, 2017.

[14] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[15] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *CVPR*, 2019.

[16] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019.

[17] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *ECCV*, 2020.

[18] Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

[19] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *CVPR*, 2020.

[20] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-relational crosstransformers for few-shot action recognition. In *CVPR*, 2021.

[21] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[22] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. *arXiv preprint*, 2019.

[23] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.

[24] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

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

[26] 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 *CVPR*, 2018.

[27] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020.

[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[30] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

[31] 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 *ECCV*, 2016.

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[33] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.

[34] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 2020.

[35] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *ECCV*, 2020.

[36] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020.

[37] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip HS Torr, and Piotr Koniusz. Few-shot action recognition with permutation-invariant attention. In *ECCV*, 2020.

[38] Li Zhang, Dan Xu, Anurag Arnab, and Philip HS Torr. Dynamic graph message passing networks. In *CVPR*, 2020.

[39] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.

[40] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *ECCV*, 2018.

[41] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.