TA²N: Two-Stage Action Alignment Network for Few-shot Action Recognition

Shuyuan Li[1*], **Huabin Liu**[1*], **Rui Qian**[1], **Yuxi Li**[1], **John See**[2]

Mengjuan Fei[3], **Xiaoyuan Yu**[3], **Weiyao Lin**[1†]

[1] Shanghai Jiao Tong University, Shanghai, China
[2] Heriot-Watt University Malaysia, Putrajaya, Malaysia
[3] Huawei Cloud, China
{shuyuanli,huabinliu,qrui9911,lyxok1}@sjtu.edu.cn, J.See@hw.ac.uk
{feimengjuan1,yuxiaoyuan}@huawei.com, wylin@sjtu.edu.cn

Abstract

Few-shot action recognition aims to recognize novel action classes (query) using just a few samples (support). The majority of current approaches follow the metric learning paradigm, which learns to compare the similarity between videos. Recently, it has been observed that directly measuring this similarity is not ideal since different action instances may show distinctive temporal distribution, resulting in severe misalignment issues across query and support videos. In this paper, we arrest this problem from two distinct aspects – action duration misalignment and action evolution misalignment. We address them sequentially through a Two-stage Action Alignment Network (TA²N). The first stage locates the action by learning a temporal affine transform, which warps each video feature to its action duration while dismissing the action-irrelevant feature (e.g. background). Next, the second stage coordinates query feature to match the spatial-temporal action evolution of support by performing temporally rearrange and spatially offset prediction. Extensive experiments on benchmark datasets show the potential of the proposed method in achieving state-of-the-art performance for few-shot action recognition.
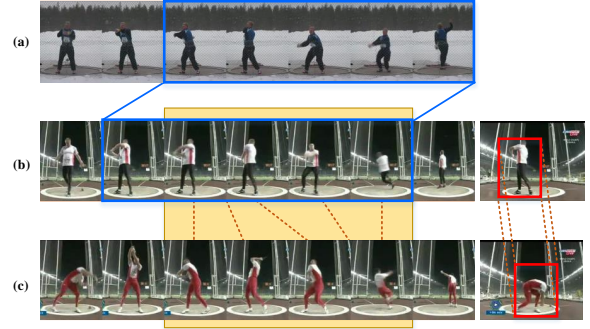
Figure 1: Example of action misalignment. (a)&(b): *action duration misalignment*. The action duration is highlighted with blue rectangles. (b)&(c): *action evolution misalignment* in temporal (left) and spatial (right) aspects. The red dashed connected lines indicate pairs of temporal or spatial positions are consistent in action content. The action category of these presented videos is *'Hammer throw'*.

Introduction

Action recognition (Tran et al. 2015; Carreira and Zisserman 2017; Wang et al. 2016) has received considerable attention in the computer vision community due to the increasing demand for video analysis in real-world scenarios. In recent years, deep learning methods have dominated the field of video action recognition with convolutional neural networks (CNNs). Numerous labeled data empower these CNNs-based methods to train a discriminative classifier for a finite set of classes. Nevertheless, in a real sense, the number of novel action categories may be limited. This problem is compounded by the laborious and expensive task of annotating all available videos for these novel categories. Consequently, the few-shot learning (FSL) task, which aims to recognize novel visual categories from very few labeled examples, has come into prominence in recent years. The solutions for image-based few-shot learning fall into three general categories: metric learning (Snell, Swersky, and Zemel

2017; Sung et al. 2018; Vinyals et al. 2016), data augmentation (Wang et al. 2018; Hariharan and Girshick 2017), and optimization-based methods (Ravi and Larochelle 2016; Wang et al. 2019). Each of them has made impressive progress in general image recognition. However, fewer studies have been carried out on few-shot video action recognition. The majority of existing approaches in this area follow the metric learning-based paradigm (Zhu and Yang 2018; Ben-Ari et al. 2020; Bishay, Zoumpourlis, and Patras 2019), which learns to compare the similarity between the videos from known classes and videos from novel classes. Recently, some research works (Bishay, Zoumpourlis, and Patras 2019; Zhang et al. 2020; Cao et al. 2020) observed that it is challenging to directly measure the similarity between videos due to the fact that different action instances show distinctive temporal distributions, *e.g.*, different temporal locations or evolution processes, along the timeline in videos, can result in severe misalignment issues between the query and support videos. Some methods attempted to address this by performing temporal alignment, *e.g.*, TARN (Bishay, Zoumpourlis, and Patras 2019) proposed a segment-by-segment attention module to perform temporal alignment at

---

*Equal contribution.

†Corresponding author.

feature level; ARN (Zhang et al. 2020) designed attention mechanisms to locate the discriminative temporal blocks. In contrast to these works, OTAM (Cao et al. 2020) explicitly aligns video sequences with a variant of the dynamic time warping algorithm. Aligning the semantic content in videos is still challenging since there exists wide variation of action instance. As such, the problem of video alignment in few-shot action recognition remains quite under-explored.

In this paper, we delve into this specific problem in few-shot action recognition from two distinct aspects – both indicative of distinct misalignment issues. First, the relative temporal location of an action is usually inconsistent between videos due to different start time and duration (as shown in Fig. 1(a)&(b)); in this paper, we define the issue of location inconsistency as ***action duration misalignment*** (ADM). Second, since action often evolves in a non-linear manner, the discriminative temporal-spatial part within the action process varies from action instances (as shown in Fig. 1(b)&(c), left and right respectively), even though they share the same semantic category and duration. We define this internal spatial-temporal malposition among action instances as ***action evolution misalignment*** (AEM).

To cope with these two types of misalignment, we devise a Two-stage Action Alignment Network (TA^{2}N) for few-shot action recognition. The first stage utilizes a Temporal Transformation Module (TTM) to predict temporal warp parameters for input video and then perform an affine transformation on the feature sequence, aligning it with the action duration period. In the second step, an Action Coordinate Module (ACM) is adopted to align the action evolution from both temporal and spatial aspects. For temporal evolution, similar motion patterns across action processes should be aggregated into the same temporal location. Thus, in temporal coordination (TC), we model the motion correlation between paired support and query videos, and then the query features could be temporally rearranged to match the support one according to the highly correlated positions. As for spatial coordination (SC), the action-specific regions (*e.g.* actors) of paired frames are required to be spatially consistent in position. Accordingly, we predict spatial offset for paired frames and then perform the corresponding movement over spatial features to align them. Benefit from our proposed two-step and coarse-to-fine align strategy, actions could be well aligned in duration and evolution, advocating a more accurate metric learning and classification. The detailed pipeline of our proposed method is illustrated in Fig. 4.

In summary, our main contributions are as follows:

- We delve specifically into the misalignment problem in few-shot action recognition, revealing and quantifying two critical aspects of this issue: the action duration and evolution misalignment.

- We propose a novel two-stage action alignment network (TA^{2}N), which performs a jointly spatial-temporal action alignment over videos, to address these two aspects of misalignment sequentially.

- Extensive experiments show that our proposed method could relieve the misalignment and achieve state-of-the-art results in few-shot video action recognition.

Related Work

Few-shot Learning A primary challenge faced in FSL is the insufficiency of data in novel classes. The direct approach to address this is to enlarge the sample size by data augmentation. Some approaches (Wang et al. 2018; Hariharan and Girshick 2017) were proposed to generate unseen data with labels to enrich the feature spaces of novel classes. Autoaugment (Cubuk et al. 2019) further automatically learns the augmentation policy to improve the generalization on various few-shot datasets. Besides, learning metrics to compare the seen and novel classes is another popular way of handling FSL. Matching network (Vinyals et al. 2016) is an end-to-end trainable kNN model using cosine as the metric, with an attention mechanism over a learned embedding of the labeled samples to predict the categories of the unlabeled data. The Prototypical Network (Snell, Swersky, and Zemel 2017) uses a feed-forward neural network to embed the task examples and perform nearest neighbor classification with the class centroids. Relation Net (Sung et al. 2018) proposed a novel network which concatenates the feature maps of two images, and proceeds to send the concatenated features to a relation net to learn the similarity. While these methods perform well on image recognition tasks, it is less optimal to transfer them directly to action recognition.

Action Recognition The state-of-the-art action recognition methods are focused on designing architectures with temporal modeling in mind. C3D (Tran et al. 2015) and I3D (Carreira and Zisserman 2017) are the most representative networks that extend VGGNet and InceptionNet respectively to 3D versions for extracting temporal information from videos. However, they lead to expensive computational costs and memory demand. Therefore, recent research has paid more attention towards efficient models such as P3D (Qiu, Yao, and Mei 2017) and R(2+1)D (Tran et al. 2018). These models decompose the 3D convolution into a 2D convolution and a 1D convolution to learn the spatial and temporal information separately.

Few-shot Action Recognition Early studies on few-shot action recognition could be traced back to CMN (Zhu and Yang 2018), which proposed a compound memory network to store matrix representations and features can be easily retrieved and updated in an efficient way. The majority of current studies on few-shot action recognition follow the metric learning paradigm. TAEN (Ben-Ari et al. 2020) encodes actions in videos as trajectories in a metric space by a collection of temporally ordered sub-actions, whereby FAN (Tan and Yang 2019) then condenses the video motion feature into a single dynamic image, which relieves the pressure of learning the distance metrics. Due to the various temporal location of actions in videos, directly comparing the similarity of two videos with misaligned actions may lead to a sub-optimal distance metric. To solve this issue, some approaches proposed to perform temporal alignment. TARN (Bishay, Zoumpourlis, and Patras 2019) proposed an attentive relation network to perform the temporal alignment implicitly at the video segment level. OTAM (Cao et al. 2020) explicitly aligns video sequences with a variant of the Dynamic Time Warping (DTW) algorithm. ARN (Zhang et al. 2020) generates attention masks to re-weight spa-
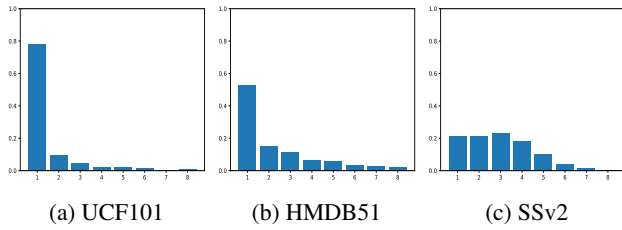
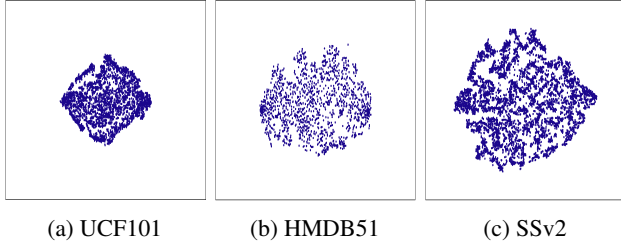Figure 2: The action start time distribution on three datasets.

Figure 3: 2-D t-SNE of the videos on three datasets.

Table 1: The estimated action evolution misalignment (AEM) score on three datasets

Dataset	UCF101	HMDB51	SSv2
Estimated AEM	0.1653	0.3697	0.6260

concentrated distribution is seen on UCF101 and HMDB51 while the distribution is slightly more scattered on the SSv2 dataset. This verifies the similar conclusion that the SSv2 faces a more serious AEM problem.

Overall, from this analysis, we can conclude that the action misalignment problem widely exists in these three datasets at varying levels. The problem is most severe on the SSv2 dataset while HMDB is more affected by this issue than the UCF101. Hence, we argue that solving the action misalignment problem is critical for few-shot action recognition, especially on the SSv2 dataset. Based on these observations, this paper seeks to address the misalignment problem by proposing a feasible framework.

Methods

Fig. 4(a) shows the framework of our method. In the following sections, we formally describe the few-shot action recognition problem, followed by detailed descriptions of modules in our framework. Finally, we describe how to optimize our model.

Definition

Problem set-up Following standard few-shot action recognition setting, the dataset is divided into three distinct parts: training set $\mathcal{C}_{train}$, validation set $\mathcal{C}_{val}$, and test set $\mathcal{C}_{test}$. The training set contains sufficient labeled data for each class while there exist only a few labeled samples in the test set. The validation set is only used to evaluate the model during training. Moreover, there are no overlapping categories between these three sets. Generally, few-shot action recognition aims to train a classification network that can generalize well to novel classes in the test set. In the specific N-way K-shot few-shot learning setting, each episode contains a support set $\mathcal{S}$ sampled from the training set $\mathcal{C}_{train}$. It contains $N \times K$ samples from N different classes where each class contains K support samples. Then Q samples from each class are selected to form the query set $\mathcal{Q}$ which contains $N \times Q$ samples. The goal is to classify the $N \times Q$ query samples only with the $N \times K$ support samples.

Feature embedding For each input video, we follow the sampling strategy described in TSN (Wang et al. 2016), which divides a video into T segments and then samples frames uniformly in each segment. Thus, each video is represented by a fixed-length frame sequence. Given the frame sequence $X = \{x_1, x_2, ..., x_T\}$, a feature embedding network $f(\cdot)$ takes it as input and embeds the sequence X into T frame-level features $f_X = f(X) = \{f(x_1), f(x_2), \ldots, f(x_T)\} \in \mathbb{R}^{C \times T \times H \times W}$. From this point, we will use f_s, f_q to represent the video-level feature of the support sample and query sample, respectively.

tiotemporal features. It utilizes augmentation strategies with self-supervised learning to enhance its feature encoder and attention mechanism. Among these methods, OTAM is the most related to our work.

Quantifying Temporal Misalignment

To further analyze the action misalignment, we quantify and compare these two types of misalignment on three video action datasets: (UCF101 (Soomro, Zamir, and Shah 2012), HMDB51 (Kuehne et al. 2011), SSv2 (Goyal et al. 2017)). Please refer to our supplementary for more details about the quantitative method and process. The quantitative results and analysis are discussed as follows.

For the *action duration misalignment (ADM)*, we aim to analyze the distribution of action start time in different datasets, which is presented in Fig. 2. For UCF101 and HMDB, the start time is distributed mainly over the first or second frame due to the fact that most videos are roughly trimmed. On the contrary, the start time on the SSv2 dataset is evenly distributed over the first four frames. This demonstrates that the actions on the SSv2 dataset are more likely to execute at various time periods, which could lead to misalignment in the action start and duration time.

For the *action evolution misalignment (AEM)*, we estimate a final AEM score for each dataset by calculating the similarity of action evolution among videos (refer to our supplementary for details about the estimation). The estimated AEM scores for the three datasets are listed in Tab. 1. It can be seen that all datasets suffer from the AEM problem. Similar to the ADM, the problem of AEM is the most serious on the SSv2. Among these datasets, the UCF101 has a lower severity of evolution misalignment, since it mainly consists of various types of sports, which provided more consistent action evolution owing to class homogeneity. Furthermore, we visualize the feature embedding of action evolution for all videos to illustrate the degree of evolution misalignment (refer to supplementary for details). It is apparent that a
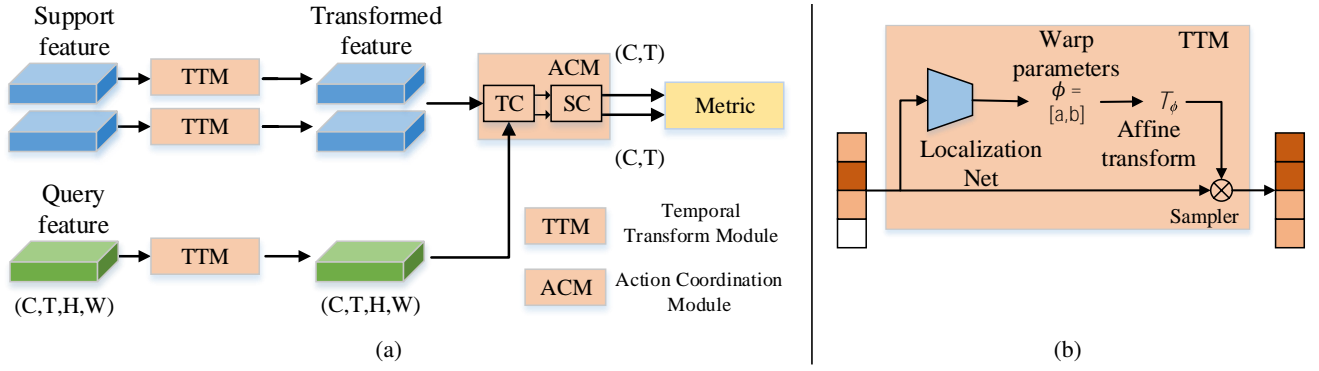
Figure 4: (a) The proposed framework. We only show 1-way for illustration. Embedded video features are first transformed by TTM (temporal transformation module) to address the action duration misalignment. Then support and query features are finely coordinated by ACM (action coordinate module) along the temporal and spatial dimensions. (b) Structure of TTM.

Temporal transform module

To address the action duration misalignment, we aim to locate the action temporally, then the duration feature could be located and emphasized while dismissing the action-irrelevant feature (e.g. background). In this way, the ADM could be eliminated. Based on this motivation, we design a Temporal Transform Module (TTM). It consists of two parts: a localization network $\mathbf{L}$ and an temporal affine transformation $\mathbf{T}$.

Specifically, given an input frame-level feature sequence f_X, the localization network generates warping parameters $\phi = (a, b) = \mathbf{L}(f_x)$ firstly. Then the input feature sequence is warped by the affine transformation $\mathbf{T}_\phi$. Succinctly, the temporal transform process is defined as:

$$\hat{f_X} = \mathbf{T}_\phi(f_X), \ \phi = \mathbf{L}(f_X) \tag{1}$$

where $\hat{f_X}$ indicates the feature sequence aligned to the action duration period, $\mathbf{L}$ consists of several light trainable layers in our implementation. Since the action duration misalignment is characteristically *linear* among frame sequences, the warping is represented using linear temporal interpolation. This also facilitates the entire pipeline differentiable and thus we can jointly train our classifier with TTM in an end-to-end manner.

The framework of TTM is illustrated in Fig. 4(b). During the episode training and testing, all the feature sequences of the support and query samples are first fed into TTM to perform first-stage temporal alignment, where their video features could be roughly aligned to their action periods. In this way, the TTM stage caters towards relieving the action duration misalignment problem.

Action coordinate module

The second type of misalignment, action evolution misalignment, results from the non-linear evolution of actions in videos, which cannot be adequately addressed by the linear-based TTM. To this end, we coordinate action evolution among videos from temporal and spatial aspects.

Temporal coordination To temporally align the action evolution among videos, similar motion patterns between videos should be aggregated to the same temporal location.
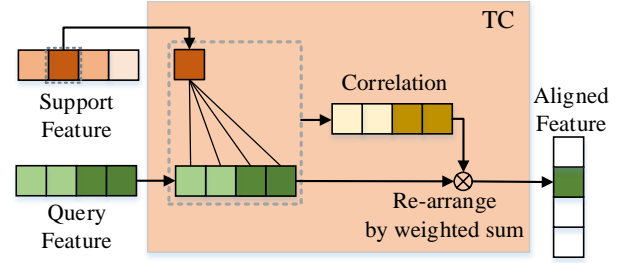

Figure 5: Temporal Coordination (TC)

We treat this as a global coordination task where the motion evolution of query video can be temporally rearranged to match the support ones. Accordingly, we model the motion evolution correlation $M \in \mathbb{R}^{T \times T}$ between support and query:

$$M = \text{Softmax}\left(\frac{(W_k \cdot G(\hat{f}_s))(W_q \cdot G(\hat{f}_q))^T}{\sqrt{dim}}\right) \tag{2}$$

where W_k, W_q indicate linear projection layer, dim is the dimension of feature $G(\hat{f})$, G is the global average pooling in spatial dimensions whose output tensor shape is $C \times T \times 1 \times 1$, *i.e.* the correlation is only calculated in the temporal dimension, Softmax limits the values in M to $[0, 1]$.

Then, we could temporally rearrange the query feature by calculating the matrix multiplication between the normalized motion correlation matrix M and the query features:

$$\tilde{f}_q = M \cdot (W_v \cdot G(\hat{f}_q)) \tag{3}$$

Similar to W_k, W_q, W_v denotes linear projection layer. In order to keep feature-space consistent, this projection are also applied to support feature $\hat{f}_s$: $\tilde{f}_s = W_v \cdot G(\hat{f}_s)$. This way, the same temporal location is in the consistent evolution and AEM in temporal aspect can be relieved. The illustration of TC is presented in Fig. 5.

Spatial coordination Temporal coordination (TC) ensures the action evolves with same process along the duration time. However, the spatial variation of actor evolution, such as the positions of actors, also being critical for action recognition, which cannot be modeled by TC. Thus,
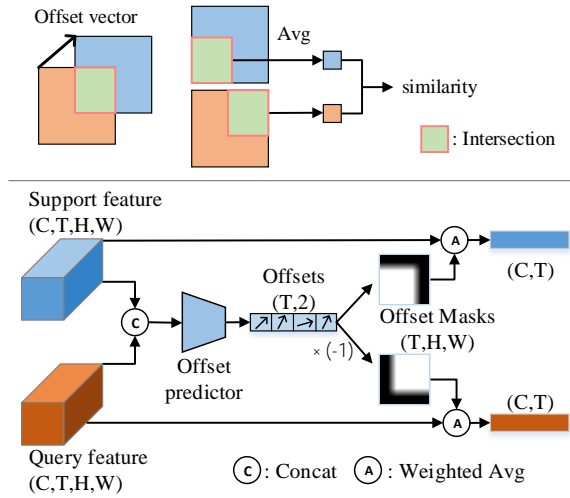
Figure 6: Spatial coordination (SC). *top*: illustration of how to measure the distance under the offset. *bottom*: details of offset prediction and mask generation.

we further devise a spatial manipulation to reduce the spatial variations in action evolution. On the basis of temporally well-aligned features, we aims to predict an spatial offset for each paired frames and then measure their similarity in the intersection area only, as shown in the top of Fig. 6.

Specifically, spatial coordination consists of two steps: light-weight offset prediction and offset mask generation. First, given the temporally well-aligned feature $\tilde{f}_s$ and $\tilde{f}_q$, they are feed into the offset predictor S to predict spatial offset $O \in \mathbb{R}^{T \times 2}$ in x and y coordinates for all timestamps:

$$O = S(\mathrm{Cat}(\tilde{f}_s, \tilde{f}_q)) \qquad (4)$$

where $\mathrm{Cat}(\cdot)$ denotes concatenation along the channel. The detailed structure of S is elaborated in our Supplementary Material. As shown in Fig.6, the predicted offset indicates the relative position vector of the action-specific region between query and support frames. Then, the intersection area can be located by its corresponding spatial offset.

To calculate the similarity in the intersection area in a differentiable way, for each frame, we use a generated offset mask I to calculate the average feature in the intersection on each feature. Moreover, the value of the mask is 1 inside the intersection area and it gradually decreases to 0 on the edge. More details about mask generation are provided in the supplementary material. Then the masks are performed over the query and support features simultaneously as the weights of weighted average:

$$\overline{f}_{s,i} = \sum_{HW} (I_{o_i} * \tilde{f}_s) / \sum I_{o_i}, \ i = 0, \dots, T \qquad (5)$$

$$\overline{f}_{q,i} = \sum_{HW} (I_{-o_i} * \tilde{f}_q) / \sum I_{-o_i}, \ i = 0, \dots, T \qquad (6)$$

where I_{o_i} is the generated mask for i-th frame.

In order to expand the exploration space of the offset predictor, we further add some perturbations on predicted offset and use the average of corresponding features of different perturbations.

Undergoing TC and SC, the action evolution misalignment among videos are dismissed in spatial-temporal aspect. The well-aligned paired features $\overline{f}_s$ and $\overline{f}_q$ are used in final distance measurement and classification as the prototypical network scheme (Snell, Swersky, and Zemel 2017).

Optimization

We train our model in similar manner as the ProtoNet (Snell, Swersky, and Zemel 2017) framework with standard softmax cross-entropy. Given the aligned feature of query sample $\overline{f}_q$, and the support prototype $\overline{p}_s^c$ for class c (obtained by applying TC to k-shot support features and average, refer to supplementary for details), we can obtain the classification probability as:

$$\mathrm{P}(x_q \in c_i) = \frac{\exp(-d(\overline{f}_q, \overline{p}_s^{c_i}))}{\sum_{c_j \in C} \exp(-d(\overline{f}_q, \overline{p}_s^{c_j}))} \qquad (7)$$

$$d(f, p) = \sum_{t=1}^{T} 1 - \frac{< f_{[t]}, p_{[t]} >}{\|f_{[t]}\|_2 \|p_{[t]}\|_2}, \qquad (8)$$

where $d(f, p)$ is the frame-wise cosine distance metric. Then the classification loss is calculated as:

$$\mathcal{L}_{cls} = - \sum_{q \in Q} \mathbb{I}(q \in c_i) \log \mathrm{P}(x_q \in c_i), \qquad (9)$$

where $\mathbb{I}$ is an indicator function. d denotes the distance metric whereby we adopt the time-wise cosine distance in our implementation. Q and C represent the query set and its corresponding collection of class label, respectively.

Experiments

Datasets and baselines

We conduct experiments on four benchmark datasets:

- *UCF101* (Soomro, Zamir, and Shah 2012): We follow the same protocol introduced in ARN (Zhang et al. 2020), where 70/10/21 classes and 9154/1421/2745 videos are included for train/val/test respectively.

- *HMDB51* (Kuehne et al. 2011): Each category contains at least 101 videos. We also follow the protocol of ARN (Zhang et al. 2020), which takes 31/10/10 action classes with 4280/1194/1292 videos for train/val/test.

- *SSv2* (Goyal et al. 2017): We adopt the same protocol as (Cao et al. 2020) where 64/12/24 classes and 77500/1925/ 2854 videos are included for train/val/test respectively.

- *Kinetics-CMN* (Zhu and Yang 2018) contains 100 classes selected from Kinetics-400, where 64/12/24 classes are split into train/val/test set with 100 videos for each class.

Competing methods We compare our method against state-of-the-art FSL action recognition methods related to temporal handling, including ProtoNet (Snell, Swersky, and Zemel 2017), CMN-J (Zhu and Yang 2020), TARN (Bishay, Zoumpourlis, and Patras 2019), ARN (Zhang et al. 2020), TRN++, and OTAM (Cao et al. 2020).

Method	Backbone	HMDB51		UCF101		SSv2		Kinetics-CMN	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CMN	ResNet-50	-	-	-	-	-	-	60.5	78.9
TARN	C3D	-	-	-	-	-	-	64.8	78.5
ARN	3D-*464*-Conv	45.5	60.6	66.3	83.1	-	-	63.7	82.4
ProtoNet	ResNet-50	54.2	68.4	74.0	89.6	33.6	43.0	64.5	77.9
TRN++	ResNet-50	-	-	-	-	38.6	48.9	68.4	82.0
OTAM*	ResNet-50	54.5	66.1	79.9	88.9	42.8	52.3	73.0	85.8
Ours	ResNet-50	**59.7**	**73.9**	**81.9**	**95.1**	**47.6**	**61.0**	72.8	**85.8**

Table 2: Few-shot action recognition results under standard 5-way k-shot settings. Note: * means our implementation.

Implementation Details

To be specific, 5-way 1-shot and 5-way 5-shot classification tasks are conducted on all datasets. For all datasets, we sample 8 frames uniformly for each video in the standard way introduced by TSN (Wang et al. 2016). Extracted frames are first resized to 256×256 and random horizontal flip is applied. Then random crop with size 224×224 is applied during training. We use the ImageNet pre-trained ResNet-50 (He et al. 2016) as the feature extractor so that we could have a fair comparison with previous methods (Cao et al. 2020). Specifically, the feature before the last average pooling layer in ResNet-50 forms the frame-level input to our TA^{2}N. During meta-training, we sample 200 episodes in a single epoch. In testing phase, we sample 5000 episodes in the meta-test spilt and report the average result. For more details about training strategies (*e.g.* optimizer, learning rate), please refer to supplementary material.

Main results

Quantitative Results The quantitative results are listed in Tab. 2. As shown in this table, our method outperforms the strong baseline ProtoNet (Snell, Swersky, and Zemel 2017) on all datasets and is competitive with state-of-the-art methods. OTAM is the current state-of-the-art method that focuses on temporal alignment (and the most related to ours). Compared to it, TA^{2}N surpasses its performance by a significant margin on most settings and datasets, demonstrating the superiority of our proposed two-stage alignment framework for action alignment. Moreover, our TA^{2}N aligns actions in the spatial-temporal aspect while OTAM only considers the temporal dimension.

Among four benchmarks, the TA^{2}N gains the most significant improvement on the SSv2. This finding also concurs with our quantitative analysis on the state of misalignment in datasets, whereby SSv2 manifests the most serious misalignment problem. This further demonstrates the effectiveness of our alignment modules. Although the UCF101 possesses a relatively less serious condition, our TA^{2}N could still improve its performance by learning a more consistent temporal feature.

Qualitative Results and Visualizations We visualize the alignment results to illustrate the effectiveness of our proposed method, which are presented in Fig. 7. It can be observed that there exists a clear duration and spatial-temporal evolution misalignment between the support and query videos. It's clear that the duration is well-aligned by

TTM	ACM		UCF101	SSv2
	TC	SC		
			74.0	40.1
✓			78.5	43.8
✓	✓		80.9	46.3
✓		✓	79.8	45.3
✓	✓	✓	**81.9**	**47.6**
	✓		78.5	44.8
	✓	✓	81.0	47.0

Table 3: Ablation of different modules, reported under 5-way 1-shot setting. ✓ means module is applied.

the TTM, which filters the insignificant background frame noise. Besides, the spatial regions coordinated by SC focus on the common action-specific part between paired frames. For example, the region of hand (row 3 col 2 in Fig. 7) and the action-specific object 'cup' (row 3 col 1 in Fig. 7) can be located precisely, which leads to a well-aligned spatial evolution for videos before being compared.

In summary, the visualization tellingly depicts the capability of TA^{2}N in correcting the misalignment.

Ablation Study

Breakdown Analysis Firstly, we break down our proposed TA^{2}N into its component parts and compare the performance gain of the TTM and ACM when applied separately. Quantitative results on UCF101 and SSv2 datasets are listed in Tab. 3. We can observe that both TTM and ACM can boost the performance of few-shot action recognition with each stage playing an equally important role in action alignment. When TTM and ACM are applied in a two-stage manner, the performance is further improved, thereby supporting our notion of a two-stage sequential design.

Further, we split the spatial and temporal coordination parts in ACM. A single TC improves the baseline with 4.5 and 4.7 in UCF and SSv2 respectively. When TTM and TC are applied sequentially, the performance grows with a large margin. This proves that our proposed TTM and TC could well address the *temporal* misalignment from two distinctive aspects. Combined with SC, we can obtain the best performance, which advocates the necessary of aligning the evolution in *spatial* dimension. In a word, the above results point towards the fact that our TA^{2}N provides an effective solution to address these two critical misalignment problems (as handled in the two stages) in few-shot action recognition task.
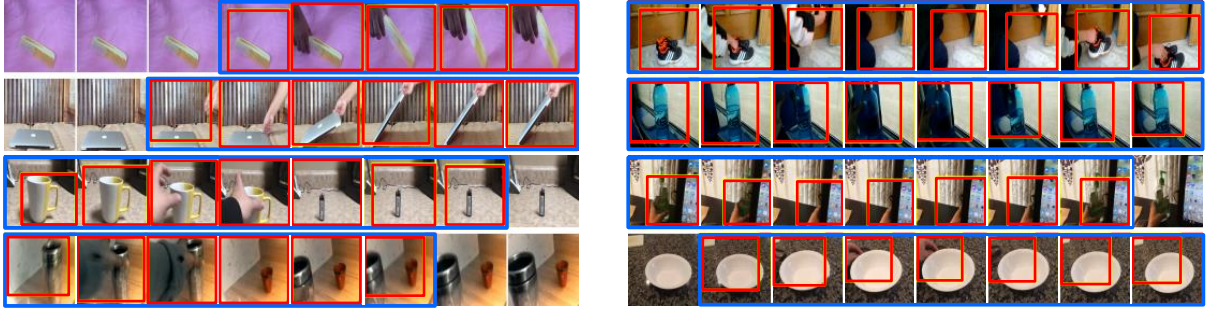
Figure 7: The visualization results on SSv2. Action duration (highlight with **_Blue_** boxes) is obtained by predicted warp parameters in TTM. Action-specific spatial region (highlight with **_Red_** box) is located with predicted offset in SC. All presented video clips are taken from the test set.
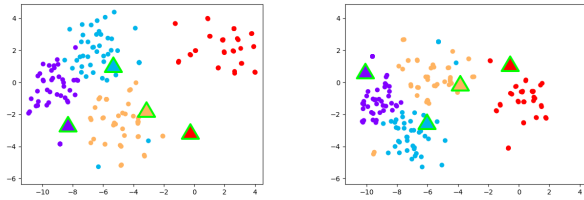

Figure 8: t-SNE feature embedding of query videos (*circles*) and support prototypes (*triangles*) on HMDB. Left/Right: before/after applying our alignment.

	HMDB51	UCF101
Enumerate	57.6	80.8
Grid-based	59.7	81.4
Mask (Ours)	**59.9**	**81.9**

Table 4: Accuracy of different designs of SC, reported under 5-way 1-shot setting.

Figure 9: Class-specific improvement using TA^{2}N compared to the Prototype baseline for SSv2.

Design of spatial coordination To reveal the effectiveness of spatial coordination (SC) module design, we compare our implementation (*Mask-based*) with other alternative designs. (1) *Enumerate*: it enumerates through all possible integer offsets in $\mathbb{Z}^2$, and straightly index the intersection area in feature x, y coordinates. The offset with the minimum metric distance between support and query is considered the optimal one. It can be regarded as a simple baseline. (2) *Grid-based*: it generates grids according to our predicted offset and then uses the re-sampling trick to sample features in the intersection area. Their results are shown in Tab. 4. As we can see, our implementation outperforms the simple enumeration by a great margin, which also demonstrates the ability of our offset predictor in SC. Compared to *Grid-based* manner, our design is more straightforward and computationally tractable, with slightly better performance.

t-SNE visualization To further illustrate the effect of our proposed TA^{2}N more intuitively, we visualize the feature embedding of query and the prototype of support before and after applying our framework by t-SNE method in Fig. 8. We can see that each cluster appears more concentrated and closer to its prototype after alignment. It further proves that our TA^{2}N can well align the query videos to support videos and obtain a more consistent feature representation.

Class-specific improvement The improvement on the SSv2 dataset for each category using the proposed TA^{2}N is presented in Fig. 9. What stands out in this figure is that the performance increases with a large margin for all categories. Moreover, some categories' accuracy (*e.g.* "pouring sth out

of sth", "approaching sth", "poking a stack of sth") rose sharply ($> 20\%$ improvement) using TA^{2}N. Interestingly, these action classes are the ones that are more vulnerable to the misalignment problem. It also prove the advantages of temporal alignment in actions with limited information.

Conclusion

This paper delves into the inevitable issue of action misalignment in few-shot action recognition and proposes a new Two-stage Action Alignment Network (TA^{2}N) to address it. Its benefits rests on two action alignment modules – the first performs temporal transformation to handle misalignment in duration, the second performs temporal rearrangement and spatially offset prediction to coordinate the evolution of action between video feature pairs. Extensive experiments affirm the effectiveness of the proposed framework.

Acknowledgements

References

Ben-Ari, R.; Shpigel, M.; Azulai, O.; Barzelay, U.; and Rotman, D. 2020. TAEN: Temporal Aware Embedding Network for Few-Shot Action Recognition. *arXiv preprint arXiv:2004.10141*.

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

Cao, K.; Ji, J.; Cao, Z.; Chang, C.-Y.; and Niebles, J. C. 2020. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10618–10627.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 113–123.

Goyal, R.; Ebrahimi Kahou, S.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*, 5842–5850.

Hariharan, B.; and Girshick, R. 2017. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, 3018–3027.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *2011 International conference on computer vision*, 2556–2563. IEEE.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, 5533–5541.

Ravi, S.; and Larochelle, H. 2016. Optimization as a model for few-shot learning.

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1199–1208.

Tan, S.; and Yang, R. 2019. Learning similarity: Feature-aligning network for few-shot action recognition. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 6450–6459.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*.

Wang, D.; Cheng, Y.; Yu, M.; Guo, X.; and Zhang, T. 2019. A hybrid approach with optimization-based and metric-based meta-learner for few-shot learning. *Neurocomputing*, 349: 202–211.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, 20–36. Springer.

Wang, Y.-X.; Girshick, R.; Hebert, M.; and Hariharan, B. 2018. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7278–7286.

Zhang, H.; Zhang, L.; Qi, X.; Li, H.; Torr, P. H.; and Koniusz, P. 2020. Few-shot action recognition with permutation-invariant attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer.

Zhu, L.; and Yang, Y. 2018. Compound memory networks for few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 751–766.

Zhu, L.; and Yang, Y. 2020. Label independent memory for semi-supervised few-shot video classification. *IEEE Annals of the History of Computing*, (01): 1–1.