Few-Shot Video Classification via Temporal Alignment

Kaidi Cao Jingwei Ji* Zhangjie Cao* Chien-Yi Chang Juan Carlos Niebles
Stanford University
{kaidicao, jingweij, caozj18, cy3, jniebles}@cs.stanford.edu

Abstract

Difficulty in collecting and annotating large-scale video data raises a growing interest in learning models which can recognize novel classes with only a few training examples. In this paper, we propose the Ordered Temporal Alignment Module (OTAM), a novel few-shot learning framework that can learn to classify a previously unseen video. While most previous work neglects long-term temporal ordering information, our proposed model explicitly leverages the temporal ordering information in video data through ordered temporal alignment. This leads to strong data-efficiency for few-shot learning. In concrete, our proposed pipeline learns a deep distance measurement of the query video with respect to novel class proxies over its alignment path. We adopt an episode-based training scheme and directly optimize the few-shot learning objective. We evaluate OTAM on two challenging real-world datasets, Kinetics and Something-Something-V2, and show that our model leads to significant improvement of few-shot video classification over a wide range of competitive baselines and outperforms state-of-the-art benchmarks by a large margin.

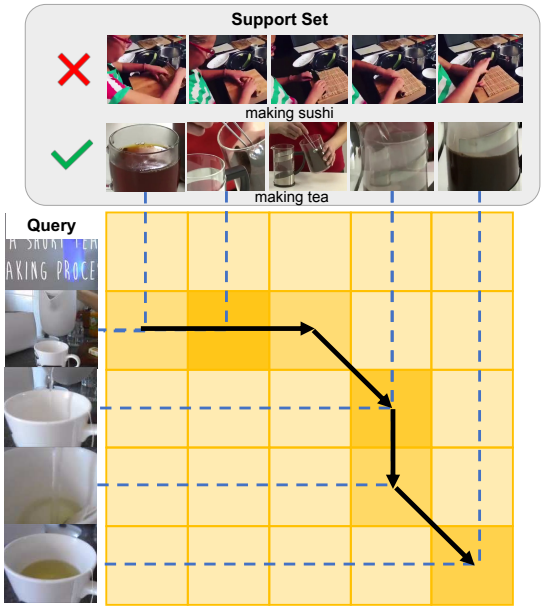

Figure 1. **Our few-shot learning approach**. We find the nearest neighbor match in the support set to make a prediction for the query video. Pairs of semantically matched frames are connected with a blue dashed line. The arrows show the direction of the ordered temporal alignment path.

1. Introduction

The emergence of deep learning has greatly advanced the frontiers of action recognition [8, 26, 41]. Currently, a major line of work focuses on learning effective representations for video classification using large amounts of labeled data [2, 22]. When a pre-trained model needs to be adapted to recognize an unseen class, typically we need to manually collect hundreds of video samples for knowledge transfer. But such a procedure is rather tedious and labor intensive. What's more, the difficulty and cost of labeling videos is much higher compared to images.

There is growing interest in learning models capable of effectively adapting themselves to recognize novel classes with only a few training examples. This is known as few-shot learning [10, 14]. Meta-learning is a promising approach for few-shot learning, the model is explicitly trained to deal with scarce training data for previously unseen classes across different episodes [12]. While the majority of recent few-shot learning work focuses on image classification, its extension to video classification is not trivial.

Videos are much more complicated than images with an additional temporal dimension. Recognizing actions such as "opening the door" requires careful modeling of temporal ordering. In the video classification literature, 3D convolution and optical flow are two main-stream methods to model short-term temporal relations [38, 41], while long-term temporal relations are usually neglected. State-of-the-art approaches commonly apply a temporal pooling module (usually mean pooling) in order to make final predictions [22, 41]. As observed in [47], averaging the deep features only captures the co-occurrence rather than the temporal or-

*Indicates equal contribution.

dering of patterns, which inevitably result in information loss.

The lack of sufficient information is even more severe when few data samples are available [19]. It is hard to learn the temporal patterns useful for few-shot classification with a limited amount of data. Utilizing the long-term temporal ordering information, which is often neglected in previous literature, is essential to few-shot learning. For example, if the model could verify that there is a procedure of pouring water before a close-up view of the just made tea, as shown in Fig. 1, the model could be confident about predicting the class of the query video as "making tea", rather than some other potential choices like "boiling water" or "serving tea". In addition, Fig. 1 also shows that for two videos in the same class, even though they both contain a procedure of pouring water followed by a close-up view of tea, the exact temporal location and duration of each atomic step can vary drastically. These non-linear temporal variations of videos pose great challenges for few-shot video classification.

With the insights above, we thus propose the Ordered Temporal Alignment Module (OTAM) for few-shot video classification, a novel temporal alignment approach that learns to estimate an ordered temporal alignment score of a query video with corresponding proxies in the support set. In concrete, we learn a matching score for each potential query-support pair by integrating segment distances only along the ordered temporal alignment path, which enforces the distance for prediction to preserve temporal ordering. Furthermore, OTAM is fully differentiable so that the model can be trained end-to-end to optimize the few-shot learning objective. This, in turn, helps the model to utilize long-term temporal information more effectively. The proposed module allows us to model the temporal evolution of videos while enabling better data efficiency in few-shot learning.

We evaluate our model on two action recognition datasets: Kinetics-400 [22] and Something-Something V2 [16]. We show that when there is only a single example available, our method outperforms the mean pooling method, which is usually adopted in current state-of-the-arts, as well as other methods tailored for few-shot video classification. We also show qualitatively that our proposed framework can learn meaningful ordered alignment paths in an end-to-end manner.

2. Related Work

Few-Shot Learning. To address few-shot learning, a direct approach is to train a model on the training set and finetune with the few data samples of the novel classes. Since the data from the novel classes is not sufficient to finetune the model with general learning techniques, several methods have been proposed to learn a good initialization model [12, 30, 33]. These works aim to relieve the difficulty of finetuning the model with limited samples. However, such methods still suffer from overfitting when training data from the novel classes is scarce. Another branch of works, which learns a common metric for both seen and novel classes, can avoid overfitting to some extent. Latent Embedding Optimization [39] employs attention kernel to measure sample distance. Relation Net [37] designs a learnable module to estimate relation score for prediction. Other methods use data augmentation to augment labeled data in the unseen classes for supervised training [17, 45]. However, video generation conditioned on categories is still an underexplored problem. Thus, in this paper, we employ a metric learning approach and design a temporal-aligned video metric for few-shot video classification.

Video Classification. A significant amount of research has tackled the problem of video classification. State of the art video classification methods have evolved from handcrafted representation learning [23, 34, 40] to deep-learning based models. C3D [38] utilizes 3D spatio-temporal convolutional filters to extract deep features from sequences of RGB frames. TSN [41] and I3D [8] uses two-stream 2D or 3D CNNs on both RGB and optical flow sequences. An issue of these video representation learning methods is their dependence on large-scale video datasets for training. Models with an excessive amount of learnable parameters tend to fail when only a small number of training samples are available.

Another concern of video representation learning is the lack of temporal relational reasoning. Making video classification sensitive to temporal ordering poses a more significant challenge to the methods above, which are tailored to capture short-term temporal features. Recently, non-local neural networks [43] introduce self-attention to aggregate temporal information in the long-term. Wang *et al*. [44] further employ space-time region graphs to perform spatio-temporal reasoning. Recently, TRN [47] proposes a temporal relational module to achieve superior performance. However, these networks inevitably pool/fuse features from different frames in the last layers to extract a single feature vector representing the whole video and fail to maintain long-term temporal ordering information. In contrast, our model can learn a video representation without the loss of temporal ordering.

Sequence Matching. Sequence matching is of great importance in the field of bioinformatics due to its application to identifying regions of similarity among different genes [3]. Instead, our focus here is on matching two video sequences [5, 9]. Inspired from text matching in machine translation [4, 42], TARN [5] utilizes attention mechanisms so as to perform temporal alignment. Nevertheless, we point out that text matching does not preserve the ordering since word ordering in different languages can vary. On the contrary, video data usually preserves the same order of atomic actions when depicting people performing certain tasks. Thus

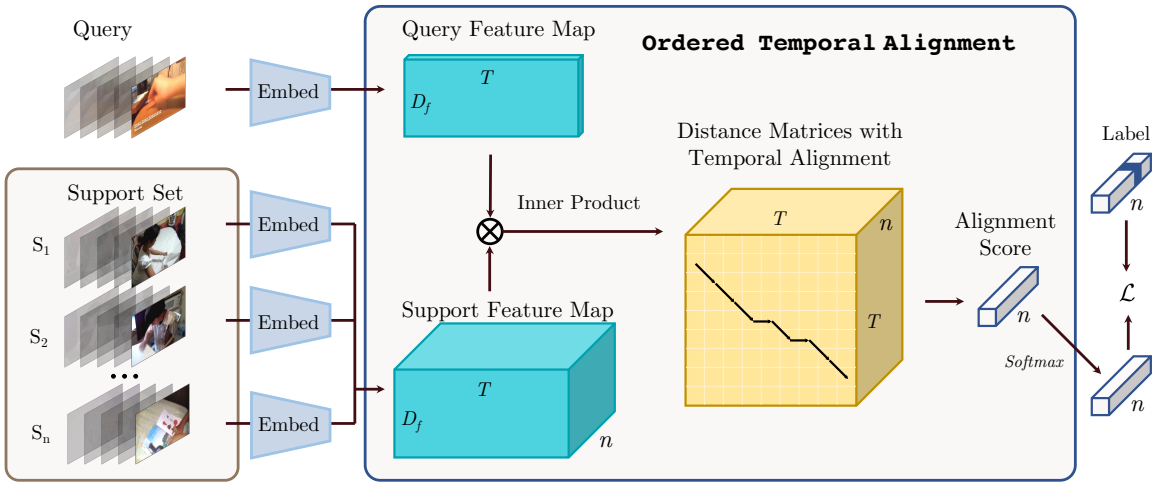

Figure 2. **Overview of our method**. We first extract per-frame deep features using the embedding network. We then compute the distance matrices between the query video and videos in the support set. Next, an alignment score is computed out of the matrix representation. Finally, we apply a softmax operator over the alignment score of each novel class.

we claim that utilizing temporal ordering could further improve data efficiency for few-shot learning in video classification tasks.

Few-Shot Learning in Videos. There are prior works exploring few-shot learning problems in the video domain. OSS-Metric Learning [24] measures OSS-Metric of video pairs to enable one-shot video classification. Mishra *et al.* [28] introduce a zero-shot method that learns a mapping function from an attribute to a class center. It has an extension to few-shot learning by integrating labeled data from the unseen classes. CMN [48] introduces a multi-saliency embedding algorithm to encode video frames into a fixed-size matrix representation. They then propose a compound memory network (CMN) to store the representation and classify videos by matching and ranking. To the best of our knowledge, the most relevant work to ours is TARN [5]. TARN also notices the importance of temporal alignment for few-shot learning and proposes to utilize attention mechanisms before measuring distances. However, including TARN, previous methods collapse the order of frames at the representation level [7, 13, 24, 25, 28, 48]. Thus, the learned models are sub-optimal for video data. In this paper, we preserve the frame ordering in video data and estimate distances with ordered temporal alignment, resulting in a more accurate final prediction.

3. Methods

Our goal is to learn a model that can classify novel classes of videos with only a few labeled examples. The wide range of intra-class spatio-temporal variations in videos poses a great challenge to few-shot video classification. We address this challenge by proposing a few-shot learning framework with the Ordered Temporal Alignment Module (OTAM). The use of OTAM sets our approach apart from previous works that fail to preserve temporal ordering and relations during the meta training and meta testing [5, 48] stages. Fig. 2 shows the outline of our model.

In the following, we first provide our problem formulation of the few-shot video classification task, then define our model and show how it can be used at the training and testing stages.

3.1. Problem Formulation

In the few-shot video classification setting, we have sufficient labeled data for classes from the training set, $\mathcal{C}_{train}$, but we only have a few labeled samples from the test set, $\mathcal{C}_{test}$. The goal of few-shot learning is to train a network that can generalize well to novel classes. Specifically, in a n-way, k-shot problem, each episode contains a support set and a query set. The support set consists of k samples each from n unseen classes, where k is usually a small integer (< 10). The algorithm then has to determine which of the support set classes each query video belongs to. Episodes are randomly drawn from a larger collection of data, which is denoted as meta sets. In our setting, we introduce 3 splits over classes as meta training $\mathcal{T}_{train}$, meta validation $\mathcal{T}_{val}$ and meta testing $\mathcal{T}_{test}$ sets.

We formulate the few-shot learning problem as a metric learning problem through learning a distance function $\phi(f_\varphi(x_1), f_\varphi(x_2))$, where x_1 and x_2 are two video samples drawn from $\mathcal{C}_{train}$ and $f_\varphi(\cdot)$ is an embedding function that maps samples to their representations. The difference between our problem formulation and the majority of previous few-shot learning approaches is that we deal with higher dimensional inputs, as they are (2+1)D volumes instead of 2D images. The extra temporal dimension in our few-shot setting demands the model to be able to learn temporal or-

dering and relations with limited data to generalize to novel classes. This poses an additional challenge that have not been properly addressed by previous works.

3.2. Model

With the problem formulation above, our goal is to learn a video distance measure by minimizing the few-shot learning objective. Our key insight is that we want to explicitly learn a distance measure independent of non-linear temporal variations by aligning the frames of two videos while preserving temporal ordering. Unlike previous works that use weighted average or mean pooling along the temporal dimension [5, 8, 38, 41, 43, 46, 48], our model can infer temporal ordering and relationships during the meta training or meta testing stage in an explicit and data-efficient manner. In this section, we breakdown our model following the pipeline illustrated in Fig. 2.

Embedding Module: The purpose of the embedding module f_φ is to generate a compact representation of a trimmed video that encapsulates its visual content. A raw video usually consists of hundreds of frames, whose information could be redundant if we were to perform per frame inference. Thus frame sampling is generally adopted as a preprocessing stage for video inputs. We follow the sparse sampling protocol described in TSN [41], which divides the video sequence into T segments and extracts a short snippet from each segment. The sparse sampling scheme allows each video sequence to be represented by a fixed number of snippets, though possibly with an unconstrained temporal variation. The sampled snippets span the whole video, enabling long-term temporal modeling.

Given a input sequence $x = \{x^1, x^2, ..., x^T\}$, we encode each snippet x^i with a CNN backbone f_φ into feature $f_\varphi(x^i)$, which results in a sequence of feature vectors $f_\varphi(x) = \{f_\varphi(x^1), f_\varphi(x^2), ..., f_\varphi(x^T)\}$. Note that the dimension of each video embedding $f_\varphi(x)$, its dimension is $T \times D_f$, rather than D_f for image embedding. We use activation before the last fully-connected layer of a CNN network as the feature embedding.

Distance Measure with OTAM:

Given two videos x_i, x_j and their embedded features $f_\varphi(x_i), f_\varphi(x_j)$, we can calculate the frame-level distance matrix $D \in \mathbb{R}^{T\times T}$ as

$$D(l,m) = 1 - \frac{f_\varphi(x_i^l) \cdot f_\varphi(x_j^m)}{||f_\varphi(x_i^l)||\ ||f_\varphi(x_j^m)||}, \tag{1}$$

where $D(l,m)$ is the frame-level distance value between the l-th frame of video x_i and the m-th frame of video x_j.

We further define $\mathcal{W} \subset \{0,1\}^{T\times T}$ to be the set of possible binary alignment matrices, where $\forall W \in \mathcal{W}$, $W_{lm} = 1$ if the l-th frame of video x_i is aligned to the m-th frame of video x_j. Our goal is to find the best alignment $W^* \in \mathcal{W}$.

$$W^* = \underset{W\in\mathcal{W}}{\operatorname{argmin}} \langle W, D\rangle, \tag{2}$$

The ideal alignment W^* would minimize the inner product between the alignment matrix W and the frame-level distance matrix D defined in Eq. (1). The video distance measure is thus given by

$$\phi(f_\varphi(x_i), f_\varphi(x_j)) = \langle W^*, D\rangle. \tag{3}$$

We propose to use a variant of the Dynamic Time Warping (DTW) algorithm [29] to solve Eq. (2). We achieve this by solving a cumulative distance function

$$\begin{aligned}\gamma(l,m) = D(l,m) \\ + \min\{\gamma(l-1,m-1), \gamma(l-1,m), \gamma(l,m-1)\}.\end{aligned} \tag{4}$$

However, in the above DTW setting, an alignment path is a contiguous set of matrix elements defining a mapping between two sequences that satisfies the following conditions: boundary conditions, continuity, and monotonicity. The boundary condition poses constraints on the alignment matrix W such that $W(1,1) = 1$ and $W(T,T) = 1$ must be true for all possible alignment paths. In our alignment formulation, though the videos are trimmed, the action in the query video does not have to match exactly its start and end with the proxy. For example, consider the action of making coffee. At the end of some videos there might be an atomic action of stirring coffee. To fit DTW to video sequences, we propose to relax the boundary condition. Instead of having a path aligning the two videos from start to end, we allow the algorithm to find a path with flexible starting and ending positions while maintaining continuity and monotonicity. To work through this, we pad two column of 0s at the start and end of the distance matrix so that it enables the alignment process to start and end at an arbitrary position. So for our method, instead of computing the alignment score on a $T \times T$ matrix, we work with the padded matrix of size $T \times (T+2)$. We further denote the indexes of the first dimension as $1, 2, ..., T$, and indexes of the second dimension as $0, 1, 2, ..., T, T+1$, for simplicity. The cumulative distance function is then changed into:

$$\gamma(l,m) = D(l,m) + \begin{cases} \min\{\gamma(l-1,m-1), \gamma(l-1,m), \gamma(l,m-1)\}, \\ \qquad\qquad m = 1 \text{ or } m = T+1 \\ \min\{\gamma(l-1,m-1), \gamma(l,m-1)\}, \text{ otherwise} \end{cases} \tag{5}$$

Note that if we follow Eq. (5) to compute the alignment score, the score by itself is automatically normalized. Since at each time step (except for $m = 0$ and $m = T+1$), the

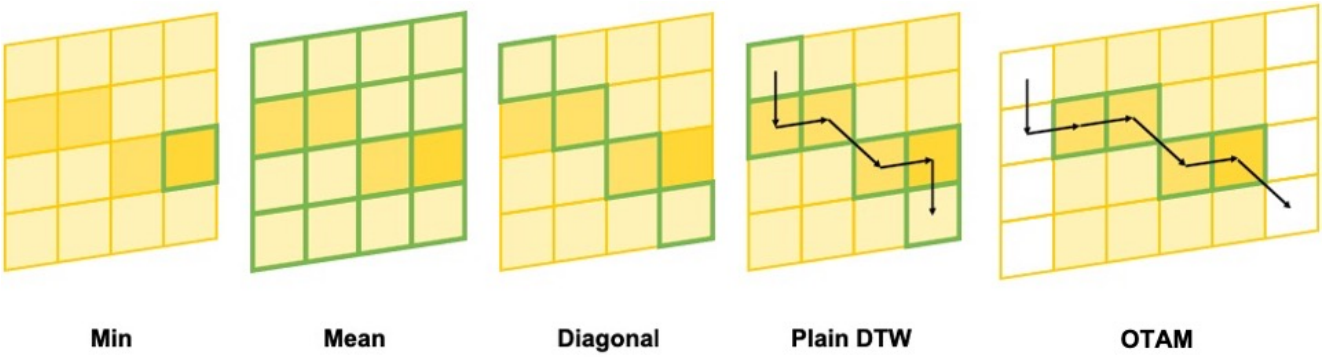

Figure 3. **Different Methods for calculating alignment score**. Each subplot shows a distance matrix. The darker of the color of an entry, the smaller the distance value is of a pair of frames. The entries with green border are the entries contributing to the final alignment score.

alignment function forces a path from $\gamma(\cdot, m-1)$ to $\gamma(\cdot, m)$, the final alignment score would be a summation of exactly T scores. In this light, alignment scores computed from different pairs of query videos and support videos are normalized to a maximum of T, which means that the scale would not be affected by the path chosen. To maintain symmetry, we repeat the same operation in the opposite direction. Our final alignment score is the average of the two alignment scores estimated under two directions.

Differentiable OTAM with Continuous Relaxation: Following recent work on continuous relaxation of discrete operation [27], we introduce a continuous relaxation to our Ordered Temporal Alignment Module. We use log-sum-exp with a smoothing parameter $\lambda > 0$ to approximate the non-differentiable minimum operator in Eq. (5)

$$\min(t_1, t_2, ..., t_n) \approx -\lambda \log \sum_{i=1}^{n} e^{-t_i/\lambda} \text{ if } \lambda \to 0. \quad (6)$$

While the use of continuous relaxation in Eq. (6) does not convexify the objective function, it helps the optimization process by allowing smooth gradients to be backpropagated through OTAM.

Training and Inference: We have shown how to compute the cumulative distance function γ and use continuous relaxation to make the objective optimizable given a pair of input videos (x_i, x_j). The video distance measure is provided by

$$\phi(f_\varphi(x_i), f_\varphi(x_j)) = \gamma(T, T+1). \quad (7)$$

In the training stage, given ground-truth video pair $(x, \hat{x})$ and support set $\mathcal{S}$, we train our entire model end-to-end by directly minimizing the loss function

$$\mathcal{L} = -\log \frac{\exp(-\phi(f_\varphi(x), f_\varphi(\hat{x})))}{\sum_{s\in\mathcal{S}} \exp(-\phi(f_\varphi(x), f_\varphi(s)))}. \quad (8)$$

To summarize, the training process can be regarded as a distance metric learning process, but all the learnable parameters are in the feature embedding module, with a fixed metric applied.

At test time we are given an unseen query video q and its support set $\mathcal{S}$, and our goal is to find the video $s^* \in \mathcal{S}$ that minimizes the video distance function

$$s^* = \underset{s\in\mathcal{S}}{\operatorname{argmin}}\ \phi(q, s). \quad (9)$$

When $k > 1$, the alignment score of the query video to each class of the support set, is the average of the sample alignment scores of that class.

4. Experiments

Our task of interest is few-shot video classification, where the objective is to classify novel classes with only a few examples from the support set. In this section, we evaluate our approach on two datasets and compare with a wide range of baselines.

4.1. Datasets

As pointed out by [46, 47], existing action recognition datasets can be roughly classified into two groups: YouTube type videos: UCF101 [36], Sports 1M [21], Kinetics [22], and crowd-sourced videos: Jester[1], Charades [35], Something-Something V1&V2 [16], in which the videos are collected by asking the crowd-source workers to record themselves performing instructed activities. Crowd-sourced videos usually focus more on modeling the temporal relationships, since visual contents among different classes are more similar than those of YouTube type videos. To demonstrate the effectiveness of our approach on these two groups of video data, we evaluate our few-shot evaluation on two action recognition datasets, Kinetics [22] and Something-Something V2 [16].

Kinetics [22] and Something-Something V2 [16] are constructed to serve as standard action recognition datasets, so we have to build their few-shot versions. For the Kinetics dataset, we follow the same split as CMN [48] and sam-

ple 64 classes for meta training, 12 classes for validation, and 24 classes for meta testing. Since there is no existing split for few-shot classification on Something-Something V2, we construct a few-shot dataset following the same rule as CMN [48]. We randomly selected 100 classes from the whole dataset. The 100 classes are then split into 64, 12, and 24 classes as the meta-training, meta-validation, and meta-testing set, respectively.

4.2. Implementation Details

For a n-way, k-shot test setting, we randomly sample n classes with each class containing k examples as the support set. We construct the query set to have n examples, where each unlabeled sample in the query set belongs to one of the n classes in the support set. Thus each episode has a total of $n(k+1)$ examples. We report the mean accuracy by randomly sampling 10,000 episodes in the experiments.

We follow the video preprocessing procedure introduced in TSN [41]. During training, we first resize each frame in the video to 256×256 and then randomly crop a 224×224 region from the video clip. We sparsely and uniformly sample $T = 8$ segments per video. For inference, we change the random crop to center crop. For the Kinetics dataset, we randomly apply horizontal flip during training. Since the label in Something-Something V2 dataset incorporates concepts of left and right, e.g., pulling something from left to right and pulling something from right to left, we do not use horizontal flip for this dataset.

Following the experiment settings from CMN, we use ResNet-50 [18] as the backbone network for TSN. We initialize the network using pre-trained weights on ImageNet [11]. We optimize our model with SGD [6], with a starting learning rate of 0.001 and decaying every 30 epochs by 0.1. We use the meta-validation set to tune the parameters and stop the training process when the accuracy of the meta-validation set is about to decrease. We implemented our framework with PyTorch [31]. The full model trains for 10 hours on 4 TITAN Xp GPUs.

4.3. Evaluating Few-Shot Learning

We compare our method with the two following categories of baselines:

4.3.1 Use Pretrained Weights

For baselines that use ImageNet pretrained weights, we follow the same setting as described in CMN. Since previous few-shot learning algorithms are all designed to deal with images, they usually take image-level features encoded by some backbone networks as inputs. To circumvent this discrepancy, we first feed frames of a video to a ResNet-50 network pretrained on ImageNet, and then average frame-level features to obtain a video-level feature. The video-level feature is then used as the input of these few-shot baselines.

Table 1. **Few-shot video classification results.** We report 5-way video classification accuracy on meta-testing set. Our approach surpasses previous state-of-the-arts by a large margin.

	Kinetics		Something V2	
Method	1-shot	5-shot	1-shot	5-shot
Matching Net [48]	53.3	74.6	-	-
MAML [48]	54.2	75.3	-	-
CMN [48]	60.5	78.9	-	-
TARN [5]	64.8	78.5	-	-
TSN++	64.5	77.9	33.6	43.0
CMN++	65.4	78.8	34.4	43.8
TRN++	68.4	82.0	38.6	48.9
OTAM (ours)	**73.0**	**85.8**	**42.8**	**52.3**

Matching Net [39]: We use an FCE classification layer in the original paper without finetuning in all experiments. The FCE module uses a bidirectional-LSTM, and each training example could be viewed as an embedding of all the other examples.

MAML [12]: Given the video-level feature as the input, we train the model following the default hyper-parameter and other settings described in [12].

CMN [48] is specially designed for few-shot video classification, it could handle video feature inputs directly. The encoded feature sequence is first fed into a multi-saliency embedding function to get a video-level feature. Final few-shot prediction is done by a compound memory structure similar to [20].

TARN [5] includes an embedding module for encoding video samples, a relation module that utilizes attention to perform temporal alignment and a deep network to learn deep distance measure on the aligned representations.

4.3.2 Finetune the Backbone

As discovered by [10, 15, 32], using cosine distances between the input feature and the trainable proxy of each class could explicitly reduce intra-class variations among features during training. Extensive experiments in [10] have shown that the Baseline++ model is competitive compared with other few-shot learning methods. So in this finetuned setting, we adapt several previous approaches with the structure of Baseline++ to serve as more competitive baselines.

TSN++: For the TSN++ baseline, we also use episode-based training to simulate the few-shot setting at the meta-train stage to optimize for generalization to unseen novel classes directly. To get a video-level representation, we average over the temporal dimension of extracted per-frame features for both query sets and support sets. The video level feature from the support set could then serve as prox-

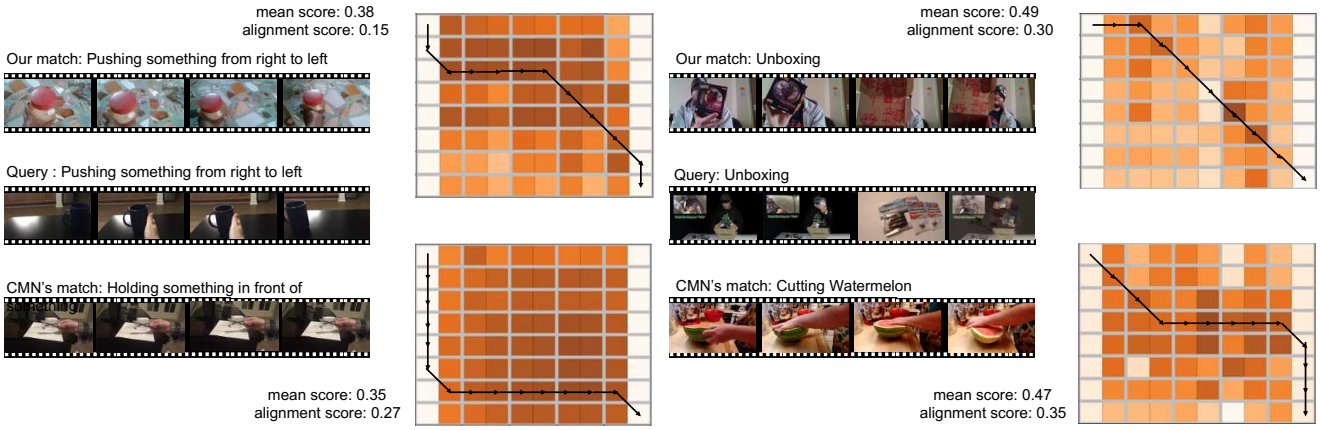

Figure 4. **Visualization of our learned score maps.** Comparison of our matched with CMN's matched results in an episode. Although the averaged score is quite high given the false matching of the query image, our algorithm is able to find the correct alignment path that minimizes the alignment score, which ultimately results in a correct prediction.

ies for each novel class. We can then obtain the prediction probability for each class by normalizing these cosine distance values with a softmax function. For inference during the meta-testing stage, we first forward each video in the support set to get proxies for each class. Given the proxies, we can then predict videos in the query set.

CMN++: We follow the setting of CMN and reimplement this method by ourselves. The only difference between CMN++ and CMN is that we replace the ImageNet pretrained feature with the feature extracted by TSN++.

TRN++: We also compare our approach against methods that attempt to learn a compact video-level representation given a sequence of image features. TRN [47] proposes a temporal relation module, which uses multilayer perceptrons to fuse features of different frames. We refer TRN++ to one of the baselines by replacing the average consensus module in TSN++ with a temporal relation module.

4.3.3 Quantitative Results

By default, we conduct 5-way few-shot classification. The 1-shot and 5-shot video classification results on both the Kinetics and Something-Something V2 datasets are listed in Tab. 1. It can be concluded that our approach significantly outperforms all the baselines on both datasets. In CMN [48], the experimental observations show that finetuning the backbone module on the meta-training set does not improve few-shot video classification performance. In contradiction, we find that with proper data augmentation and training strategy, a model could be trained to generalize better on unseen classes in a new domain given the meta-training set. By comparing the results of TSN++ and TRN++, we could conclude that considering temporal relation explicitly helps with model generalization on unseen classes. Compare to TSN++, the improvement brought by CMN++ is not as large as the gap on ImageNet pretrained features reported in the original paper. This may be caused by our use of a more suitable distance function (cosine distance) during meta-training so that the frame-level feature is more discriminative among unseen classes. Finally, note that OTAM outperforms all the finetuned baselines by a large margin. This demonstrates the importance of considering temporal ordering information when dealing with few-shot video classification problems.

4.3.4 Qualitative Results and Visualizations

We show qualitative results comparing CMN and OTAM in Fig. 4. In particular, we observe that CMN has difficulty in differentiating two actions from different classes with very similar visual clues among the frames, e.g., backgrounds, as can be seen from the distance matrices in Fig. 4. Though our method cannot alter the fact that the two visually similar action clips would have an averagely lower frame-wise distance score, it is able to find a temporal alignment path that minimizes the cumulative distance score between the query video and the true support class video. Though the mean score of OTAM's prediction is lower than the match of CMN's, out method succeeds in making the right prediction by calculating a lower alignment score out of the distance matrix.

4.4. Ablation Experiments

Here we perform ablation experiments to demonstrate the effectiveness of our selections of the proposed method. We have shown in Section 4.3 that explicitly modeling the temporal ordering plays a vital role for generalization to unseen classes. We now analyze the effect of different temporal alignment approaches.

While having the cosine distance matrix D, there are several choices we could adopt to extract the alignment score out of the matrix, as visualized in Fig. 3. In addition to our

Table 2. **Temporal matching ablation study.** We compare our method with temporal-agnostic baselines and the Plain DTW approach.

	Kinetics		Something V2	
matching type	1-shot	5-shot	1-shot	5-shot
Min	52.4	71.6	29.7	38.5
Mean	67.8	78.9	35.2	45.3
Diagonal	66.2	79.3	38.3	48.7
Plain DTW	69.2	80.6	39.6	49.0
OTAM(Ours)	**73.0**	**85.8**	**42.8**	**52.3**

proposed method, we consider several heuristics for generating the scores. The first is "Min", where we use the smallest element in the matrix D to represent the video distance value. The second is "Mean", for which we average over the cosine distance value of all pairs of frames. These two choices neglect the temporal ordering. We then introduce a few potential choices that explicitly consider sequence ordering when computing the temporal alignment score. A direct scheme is to take an average over the diagonal of the distance matrix. The assumption behind this approach is that the query video sequence is perfectly aligned with its corresponding support proxy of the same class, which could be unrealistic for real-world applications. To allow for a more adaptive alignment strategy, we introduce Plain DTW and OTAM. Here, Plain DTW in Tab. 2 means that there is no padding so that W_{11} and W_{TT} are assumed to be in the alignment path, and for each time step during computing alignment score we allow a possible movement choice among $\longrightarrow$, $\searrow$ and $\downarrow$.

The results are shown in Tab. 2. It shows that we can improve few-shot learning by considering temporal ordering explicitly. There are some slight differences in performance between the Diagonal and Mean methods in two datasets here. There are fewer visual clues in each frame of Something-Something V2 than that of Kinetics, so the improvement of Diagonal is more prominent on Something-Something V2. At the same time, the gap is small for Kinetics. However, we see that through adaptive temporal alignment, our method consistently improves the baselines on two datasets by more than 3% across 1-shot and 5-shots. This shows that by reinforcing the model to learn an adaptive alignment path across query videos and proxies, the final model could learn to encode better representations for the video, as well as a more accurate alignment score.

The next ablation study is on the sensitivity of the smoothing parameter λ. Intuitively, a smaller λ functions more like the min operation, and a larger λ means a heavier smoothing effect over the values in nearby positions. We experimented on λ within the value set of $[0.01, 0.05, 0.1, 0.5, 1]$.

The results are shown in Fig. 5. In general, the performance is stable across different values of λ. We observe

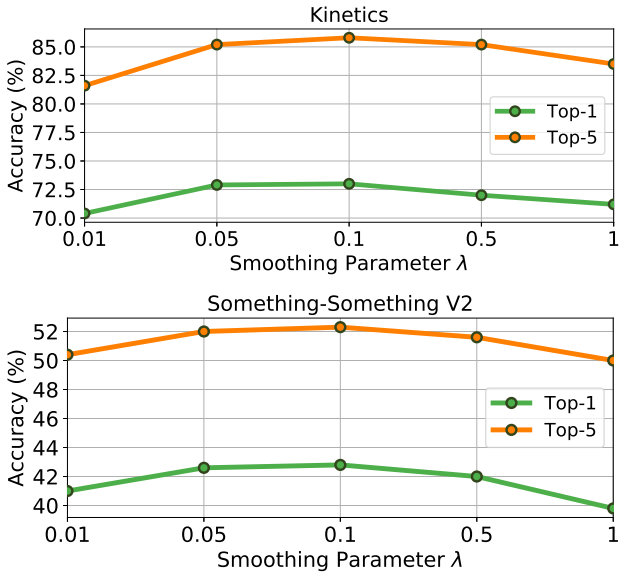

Figure 5. **Smoothing factor sensitivity.** We compare the effect of using different smoothing factors. We notice that a suit-able λ is essential for the representation learning. And in general, the perforamance is stable across different values of λ.

that in practice, λ ranges from 0.05 to 0.1 works relatively good under the setting of both two datasets. Thus we notice that a suitable λ is essential for the representation learning. When λ is too small, though it can function most similarly as the real min operator, the gradient is too imbalanced so that some pairs of frames are not adequately trained. On the contrary, a large λ might be too smooth so that the difference among all kinds of alignments is not notable enough.

5. Conclusion

We propose Ordered Temporal Alignment Module (OTAM), a novel few-shot framework that can explicitly learn distance measure and representation independent of non-linear temporal variations in videos using very few data. In contrast to previous works, OTAM dynamically aligns two video sequences while preserving the temporal ordering, and it is directly optimized for the few-shot learning objective in an end-to-end fashion. Our results and ablations show that our model significantly outperforms a wide range of competitive baselines and achieves a state of the art results on two challenging real-world datasets. A future direction is to study more interpretable few-shot video classification algorithms.

Acknowledgements This work has been partially supported by JD.com American Technologies Corporation (JD) under the SAILJD AI Research Initiative. This article solely reflects the opinions and conclusions of its authors and not JD or any entity associated with JD.com.

References

[1] The 20bn-jester dataset v1. https://20bn.com/datasets/jester. 5

[2] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 1

[3] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997. 2

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[5] M. Bishay, G. Zoumpourlis, and I. Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv preprint arXiv:1907.09021*, 2019. 2, 3, 4, 6

[6] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 6

[7] C. Careaga, B. Hutchinson, N. Hodas, and L. Phillips. Metric-based few-shot learning for video action recognition. *arXiv preprint arXiv:1909.09602*, 2019. 3

[8] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 2, 4

[9] C.-Y. Chang, D.-A. Huang, Y. Sui, L. Fei-Fei, and J. C. Niebles. D^3TW : Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. *arXiv preprint arXiv:1901.02598*, 2019. 2

[10] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 1, 6

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 6

[12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 1, 2, 6

[13] Y. Fu, C. Wang, Y. Fu, Y.-X. Wang, C. Bai, X. Xue, and Y.-G. Jiang. Embodied one-shot video recognition: Learning from actions of a virtual embodied agent. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 411–419, 2019. 3

[14] V. Garcia and J. Bruna. Few-shot learning with graph neural networks. In *ICLR*, 2017. 1

[15] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 6

[16] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, volume 2, page 8, 2017. 2, 5

[17] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017. 2

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[19] J. Ji, K. Cao, and J. C. Niebles. Learning temporal action proposals with fewer labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7073–7082, 2019. 2

[20] Ł. Kaiser, O. Nachum, A. Roy, and S. Bengio. Learning to remember rare events. *arXiv preprint arXiv:1703.03129*, 2017. 6

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 5

[22] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 5

[23] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008. 2

[24] O. Kliper-Gross, T. Hassner, and L. Wolf. One shot similarity metric learning for action recognition. In *International Workshop on Similarity-Based Pattern Recognition*, pages 31–45. Springer, 2011. 3

[25] S. Kumar Dwivedi, V. Gupta, R. Mitra, S. Ahmed, and A. Jain. Protogan: Towards few shot learning for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[26] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019. 1

[27] A. Mensch and M. Blondel. Differentiable dynamic programming for structured prediction and attention. *ICML*, 2018. 5

[28] A. Mishra, V. K. Verma, M. S. K. Reddy, S. Arulkumar, P. Rai, and A. Mittal. A generative approach to zero-shot and few-shot action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 372–380. IEEE, 2018. 3

[29] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007. 4

[30] A. Nichol and J. Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018. 2

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 6

[32] H. Qi, M. Brown, and D. G. Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018. 6

[33] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018. 2

[34] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360. ACM, 2007. 2

[35] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 5

[36] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[37] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1, 2, 4

[39] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 2, 6

[40] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 2

[41] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 1, 2, 4, 6

[42] S. Wang and J. Jiang. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*, 2016. 2

[43] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2, 4

[44] X. Wang and A. Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–417, 2018. 2

[45] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018. 2

[46] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 4, 5

[47] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 1, 2, 5, 7

[48] L. Zhu and Y. Yang. Compound memory networks for few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–766, 2018. 3, 4, 5, 6, 7