

# Leveraging triplet loss for unsupervised action segmentation

Elena Bueno-Benito <sup>\*</sup>  
ebueno@iri.upc.edu

Biel Tura Vecino <sup>\*†</sup>  
bieltura@amazon.co.uk

Mariella Dimiccoli <sup>\*</sup>  
mdimiccoli@iri.upc.edu

<sup>\*</sup> Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain

<sup>†</sup> Amazon Alexa AI, UK, Cambridge

## Abstract

In this paper, we propose a novel fully unsupervised framework that learns action representations suitable for the action segmentation task from the single input video itself, without requiring any training data. Our method is a deep metric learning approach rooted in a shallow network with a triplet loss operating on similarity distributions and a novel triplet selection strategy that effectively models temporal and semantic priors to discover actions in the new representational space. Under these circumstances, we successfully recover temporal boundaries in the learned action representations with higher quality compared with existing unsupervised approaches. The proposed method is evaluated on two widely used benchmark datasets for the action segmentation task and it achieves competitive performance by applying a generic clustering algorithm on the learned representations. <sup>1</sup>

## 1. Introduction

Unconstrained videos capturing real-world scenarios are usually long, untrimmed and contain a variety of actions which can be effortlessly divided by a human observer into semantically homogeneous units. The task of action segmentation, which we target in this work, is the process of identifying the boundaries of an action, i.e. *pour water*, in an untrimmed video of an activity, i.e. *making tea*, even when temporally adjacent actions may have very small visual variance between them. This process is a key step in understanding and contextualizing the video. It is also crucial for video browsing, indexing, and summarisation and has applications in areas such as surveillance systems, action recognition, video content-based retrieval, assistive technologies, and robot-human interactions.

This problem has been traditionally tackled through supervised learning approaches [9, 13, 17, 18, 21, 33]. Such approaches require a large amount of annotated training data and typically suffer from domain adaptation problems,

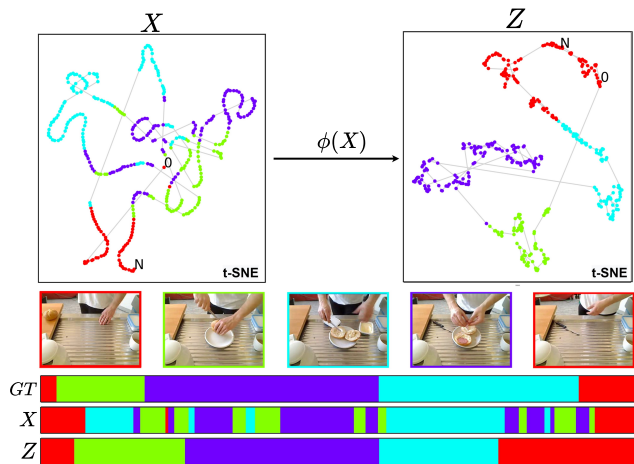


Figure 1. Our approach learns a parametrized function  $\phi$  that transforms the input feature space ( $X$ ) into a new one ( $Z$ ), where actions, visualized through different colours, can be easily unveiled by a generic clustering algorithm. The continuous line connects points representing frames from time 0 to  $N$  in a t-SNE projection. GT stands for ground truth.

being unable to generalize to large-scale target domains. More recently, weakly-supervised and semi-supervised approaches have shown to be an effective way to learn video representations convenient for action segmentation without requiring or requiring very little manual annotations [2, 7, 10, 11, 14, 19, 22–25, 29]. However, these approaches are still data-hungry and computationally expensive. Unsupervised approaches have developed following two different research lines [1, 16, 20]. Most of them focus on grouping actions across videos and rely on the use of activity labels [15, 16, 20, 28, 30, 32], therefore putting more emphasis on the quality of the representation. A few of them, the most computationally efficient, act on a single video to recover clusters [26] or detect temporal boundaries [8] and do not require any manual annotation.

Our approach stands between these two research lines. We assume that the atomic actions can effectively be modelled as clusters in an underlying representational space and we propose a novel framework that maps the initial feature space of a video into a new one, where the temporal-

<sup>\*</sup>Work done during an internship at the IRI.

<sup>1</sup><https://github.com/elenabbuenob/TSA-ActionSeg>

semantic clusters corresponding to atomic actions are unveiled. Similarly to other unsupervised approaches that rely on similar assumptions [15], our focus is on representation learning. However, similarly to [8, 26], our method takes as input a single video and doesn't care whether the same action is present in similar videos or not. This has considerable practical advantages for downstream applications since it can be in principle applied to any video no matter the dataset it belongs to nor if there exist videos having a similar temporal structure. Action segmentation is obtained by applying a generic clustering algorithm on the learned temporal-semantic aware (TSA) representations (see Figure 1).

Our contributions are as follows:

- We introduce the novel approach to action representation learning that uses a shallow network and the single video itself as input, without the need for additional training data.
- We demonstrate the effectiveness of using a triplet loss operating on similarity distributions with a novel triplet selection strategy based on a downsampled temporal-semantic similarity weighting matrix for the task of action segmentation.
- We detail ablation study and we achieve state-of-the-art metrics on the *Breakfast* and *Youtube INRIA Instructional* benchmark datasets without using any training data than a single video itself as input.

## 2. Related work

**Fully supervised approaches.** Action Segmentation has been traditionally tackled as a supervised learning problem. Existing approaches belonging to this category differ mainly in the way temporal information is taken into account by the model. Traditional approaches follow a two-step pipeline, that first generates frame-wise probabilities and then feeds them to high-level temporal models as in the Hidden Markov Model Tool Kit (HTK) approach [13] or in [33], which is based on recurrent neural networks. Lately, there has been a proliferation of models based on temporal convolutions to directly classify the video frames. Encoder-Decoder Temporal Convolutional Networks (ED-TCNs) [17] use a hierarchy of temporal convolutions to perform fine-grained action segmentation, but they can act solely on low-temporal resolution videos. Instead, Multi-Stage Temporal Convolutional Network (MS-TCN and its improved version MS-TCN++) can act on the full temporal resolution of the videos and achieves increased performance [9, 21]. Spatio-temporal convolutional layers [18] have shown promising results in capturing temporal dependencies while being easier to train than previous methods.

The main drawback of traditional supervised approaches to action segmentation is the requirement of a large amount of quality labelled data for training, which limits their applicability to large-scale domains outside of existing pre-segmented datasets [13, 21].

**Weakly and semi-supervised approaches.** To alleviate the need for large annotated datasets, weakly supervised techniques for video segmentation involve using transcripts (ordered list of the actions occurring in the video), visual similarities, and audio information to generate pseudo-labels for training [10]. In [14], a Gaussian Mixture Models + Convolutional Neural Networks (GMM+CNN) is first initialized and used to infer the segments of a video given a transcription of it. The new segmentation is used to re-estimate and update the model parameters until convergence. In [24], a recurrent neural network is used to model a discriminative representation of subactions, and a coarse probabilistic model to allow for temporal alignment and inference over long sequences. Some approaches use machine learning models to infer the segments of the video [22]. Other approaches, such as those based on frame-to-frame visual similarities [11], self-attentions mechanism [23] or iterative soft boundary assignment [7], enforce consistency between the video and labels without the need for temporal supervision. In the work [10], a network is trained on long videos which are only annotated by the set of present actions and are trained by dividing the videos into temporal regions that contain only one action class and are consistent with the set of annotated actions. The work in [25] adopts a Hidden Markov Model grounded on a Gated Recurrent Unit (GRU) for labelling video frames. This model has been subsequently improved in [19], where the model is trained through a Constrained Discriminative Forward Loss (CDFL) that accounts for all candidate segmentations of a training video, instead than a single one. [2] uses speech as additional sources of human-generated information in a weakly-supervised framework.

Recent work has proposed a semi-supervised approach [29] (ICC) consisting of a previous unsupervised training with a contrastive loss followed by a supervised training step with a small amount of labelled samples. These methods are limited to videos with transcripts and cannot be generalized to unconstrained videos.

**Unsupervised learning approaches.** Unsupervised learning approaches typically learn action representation in a self-supervised fashion and then apply a clustering algorithm to obtain the action segmentation (assuming that the number of clusters is known). Some methods model that minimizes prediction errors exploiting the order of scripted activities [15] or combining temporal embedding with visual encoder-decoder pipelines [30]. Other approaches use deep learning architectures, including an ensemble of autoencoders, and classification networks that exploit the relation between actions and activities [20]. Based on the assumption that in task-oriented videos actions occur in a similar temporal space, [15, 28] learn a strong temporal regularization that partially hides visual similarity. [32] propose a Self-Supervised Co-occurrence Action Parsing method

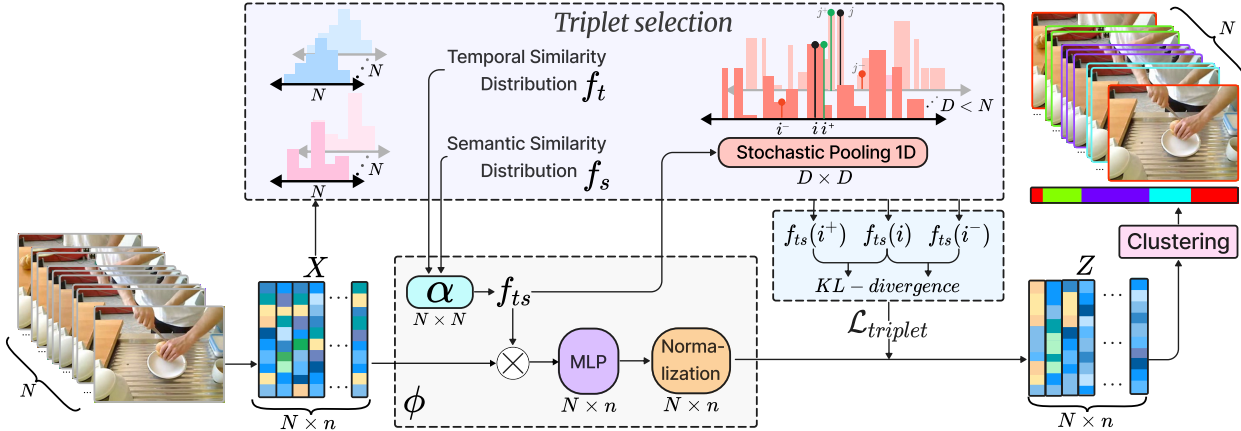


Figure 2. Overview of the proposed TSA framework illustrated on a sample video of the Breakfast Dataset: network architecture transforming the initial features  $X$  into the learned features  $Z$  through a shallow network with a novel triplet selection strategy and a triplet loss based on similarity distributions.

(SSCAP), for unsupervised temporal action segmentation which takes the recurrence of sub-actions into account in estimating the temporal path, and hence is able to handle complex structures of activities. Recently, [16] proposed a joint self-supervised representation learning and online clustering approach, which uses video frame clustering as a pretext task and hence directly optimizes for unsupervised activity segmentation (TOT+TCL).

Even if these approaches do not require labelled data, they are data-hungry and are not suitable for transferring the learned knowledge to a dataset with a different distribution and they demonstrated modest performance for the task of action segmentation at the video level. In contrast, we aim at unveiling the clusters underlying a single video. There is limited work in the literature on learning action representations in a self-supervised manner within a single video. [1] proposes a model based on an encoder LSTM architecture with Adaptive Learning (LSTM+AL) that minimizes the prediction error of future frames and assigns segmentation boundaries based on the prediction error of the next frame. Some recent works proposed to learn event representations [4, 5] and the underlying graph structure from a single sequence (DGE) [6], where temporal windows are taken into account instead of all frames. In both cases, these approaches were not tested for the action segmentation of complex activity videos (high-temporal resolution), but only on low-temporal resolution image sequences.

**Fully unsupervised approaches.** Clustering methods, which generate a partition of the input data based on a specific similarity metric, have been poorly investigated within the field of action segmentation. However, very recent work [26] has shown that simple clustering approaches, i.e.  $K$ -means, are instead a strong baseline for action segmentation. They hence proposed a new clustering approach called Temporally-Weighted FINCH (TW-FINCH), which is similar in spirit to the clustering approach named FINCH [27]

but takes into account temporal proximity in addition to semantic similarity. Recently, [8] proposed to detect action boundaries (ABD) by measuring the similarity between adjacent frames based on the insight that actions have internal consistency within and external discrepancy across actions. We based our approach on the same insight that we modelled via a deep metric learning approach.

### 3. Methodology

We assume that the representational clustering grounding action segmentation encodes both temporal and semantic similarity, based on two observations: (i) an action in a video is a sequence of images temporally continuous, therefore temporal adjacent frames are likely to belong to the same action. (ii) frames corresponding to the same action (but not necessarily temporal adjacent) should have similar representation, encoding the common underlying semantic.

Formally, let  $X \in \mathbb{R}^{N \times n}$  denote the matrix of  $n$ -dimensional feature vectors for a given sequence of  $N$  frames. We aim at learning a parametric function  $\phi$  such that given the input feature matrix  $X$ , new Temporal-Semantic Aware (TSA) (see Figure 2) representations  $Z \in \mathbb{R}^{N \times n}$  are obtained as  $Z = \phi(X)$ .

#### 3.1. Triplet loss and triplet selection

To learn  $\phi$ , we minimize a triplet loss function (defined in Equation 1) that implements an original approach to select the triplets appropriately by relying on temporal-semantic similarity distributions  $f_{ts}$  obtained as the weighted sum of the temporal and the semantic similarity distributions, say  $f_t$  and  $f_s$ ,  $f_{ts} = \alpha \cdot f_t + (1 - \alpha) \cdot f_s$ , where  $\alpha \in [0, 1]^{N \times 1}$  a vector of learning parameters of the function  $\phi$ . Under the model assumptions, it is easy to see that the similarity of  $f_{ts}(k)$  and  $f_{ts}(k')$  will be large when  $k$  and  $k'$  belong to the same event, and it will be small when  $k$  and  $k'$  belong two different events.

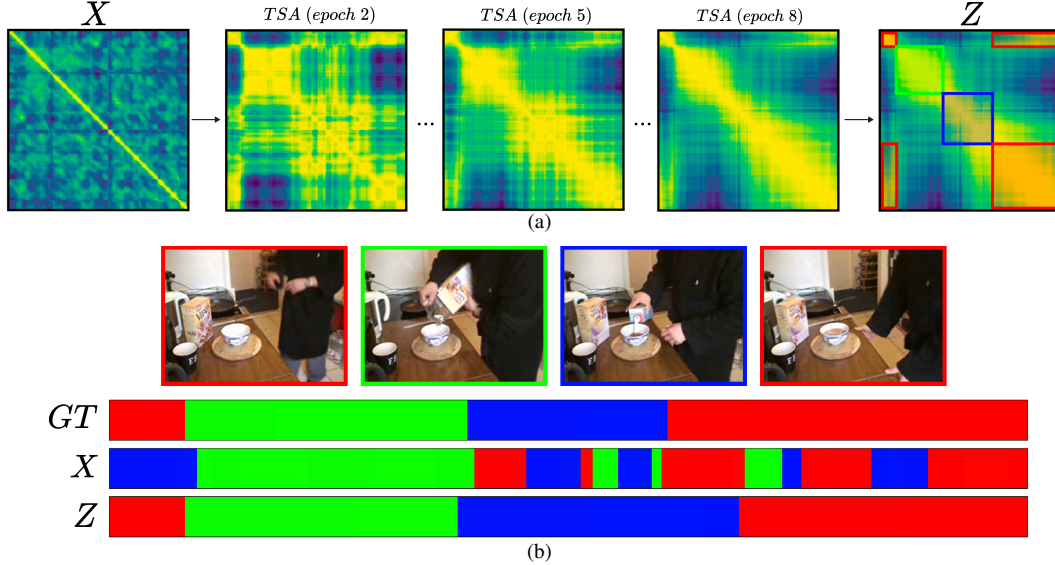


Figure 3. Example of training and result obtained by using the TSA approach for a sample video of the *Breakfast Action Dataset* (P34\_cereals). **(a)** Cosine similarity affinity matrix for initial features  $X$  and evolution of the learned features  $Z$  for different training epochs. Actions are highlighted as neighbour communities referring to the segmentation of the video when a clustering algorithm is applied. **(b)** Segmentation plots showing the ground truth, and the result of applying the same clustering algorithm to initial features  $X$  (IDT) and to the learned features  $Z$  (TSA).

**Semantic similarity distribution:** To define  $f_S$ , we assume that the set of most similar frames in the original feature space of an anchor  $i$  is very likely to be part of the same action. The similarity of an anchor  $i$  to all other frames is defined element-wise via a pairwise similarity, upon normalization to the total unit weight,  $f_S = w_{ij}/W$ , with  $W = \sum_{i,j \in E} w_{ij}$  and  $w_{ij} = \exp(-(1 - d(x_i, x_j))/h)$ , and where  $E$  is the set of pairwise relations,  $d(\cdot, \cdot)$  is the cosine distance and  $h$  is the filtering parameter of the exponential function. The resulting pairwise similarities are normalized to represent empirical joint probability distributions between pairs of elements in the sequence.

**Temporal similarity distribution:** To define  $f_t$ , we assume that as we move away from the anchor  $i$ , the likelihood of a feature vector  $x_{j \neq i}$  to represent the same action as frame  $i$  decreases. To model this behaviour, we define a weight function  $w(\cdot)$  that depends on the temporal frame distance  $d$  from the given frame as  $w(d) = -1 + 2 \exp(-\frac{1}{\beta}d)$  where  $\beta$  is a constant that controls the slope of the weight function and  $d$  is the temporal distance between frames. By imposing that  $w(L/2) = 0$ , and then solving for  $\beta$ , we get that the constant  $\beta$  can be expressed in terms of the positive window length, that is:  $\beta = -L/(2 \ln(\frac{1}{2}))$ .

The temporal and semantic distributions are downsampled to reduce computational costs using stochastic pooling during the training. An anchor index is randomly selected from the set of downsampled indices  $i \in D$ . Its set of positive samples  $\mathcal{P}_i$  is taken as the 5% of the frames with the

highest similarity values in  $i$ -row of the temporal-semantic affinity matrix  $f_{ts}$ . We define the negative set  $\mathcal{N}_i$  as the frames whose  $i$ -row  $f_{ts}$  is between the mean and the sum of the mean and standard deviation of the similarity metric. Our triplet loss is defined as:

$$\mathcal{L}_{triplet} = \frac{1}{D} \sum_{i \in D} \max(0, KL(f_{ts}(i)||f_{ts}(i^-)) - KL(f_{ts}(i)||f_{ts}(i^+))) \quad (1)$$

where  $KL$  represent the KL-divergence of the temporal-semantic similarity distribution  $f_{ts}$ . For each loss term, given an anchor index  $i \in D$  with  $D < N$ , we define the triplet  $\{i, i^+, i^-\}$  where  $i^+ \in \mathcal{P}_i$  and  $i^- \in \mathcal{N}_i$  that they are the sets of positive and negative indices, respectively.

Using probability distribution functions (PDFs),  $f_{ts}$  as feature vectors, instead of initial features  $X$  can provide several benefits for action segmentation. Since PDFs consider all the information in the video for each frame, they are smoother and more robust compared to the initial features extracted from the data  $X$  (they can be noisy or contain irrelevant information). In the ablation study (see Section 4.3), we will prove that this can improve the accuracy and robustness of the triplet loss for action segmentation.

## 4. Experimental evaluation

### 4.1. Experimental setup

**Datasets:** We report results on two widely used temporal action segmentation datasets: (1) the *Breakfast Action*

*Dataset* [12] consists of 10 activities related to breakfast preparation, performed by 52 different individuals in 18 different kitchens for a total of 1712 individual videos of 64 dimensions each feature vector; (2) the *Youtube INRIA Instructional Dataset* [2] consists on 150 instructional videos from YouTube whose feature vectors are 3000-dimensional. These include 5 different unrelated activities (changing tire, preparing coffee, performing a Cardio Pulmonary Resuscitation, jumping a car and repotting a plant), lasting 2 minutes on average. The most challenging part of this dataset is the amount of background frames present in each video, which reaches up to 63.5% of the frames in the dataset.

**Evaluation metrics:** We use a clustering algorithm on the learned representations to segment the video into its atomic actions. To match the predicted segmentation labels and the ground truth, we follow the Hungarian matching algorithm to obtain a one-to-one mapping between the predicted labels (cluster index) and the ground truth at video-level. Following previous work [1, 8, 26, 28], we report three widely used metrics: (i) accuracy of the segmentation and action identification, computed as the *Mean over Frames (MoF)* metric, which indicates the percentage of action predicted frames correctly labelled after the Hungarian matching. (ii) Similarity and diversity of the predicted segments, calculated as the *Intersection over Union (IoU)* metric, also known as the Jaccard Index or Jaccard similarity coefficient. (iii) The *F1-score* computed across the predicted segments and the known ground truth to evaluate the quality of the action segmentation. Note that, among unsupervised methods, [15, 28] look for a one-to-one global matching (shared across all videos of a given task) instead that at the video-level. This generally leads to poorer performance than computing the the Hungarian at video-level.

## 4.2. Implementation details

**Input features.** To ensure a fair comparison to state-of-the-art methods targeting the action segmentation task, we use the same datasets and input features for the frame-level initial representations as in [1, 8, 15, 26, 28, 30]. For the *Breakfast Action* we use the Improved Dense-Trajectory (IDT) [31] features. These were provided by the authors of CTE [15] in their open-sourced implementation. For *Youtube Inria Instructional* we use a set of frame-level representations given by their original authors, which are features vectors formed by a concatenation of HOF descriptors and features extracted from VGG16-conv5 network.

**Model architecture** We used a shallow neural network consisting in our case of a multi-layer perceptron with a single hidden layer followed by a ReLu activation function. This makes our approach (Figure 2) easy to train and more suited for practical applications than existing approaches consisting of multiple convolutional layers and/or recurrent networks. Empirical experiments showed that using a sin-

gle hidden layer was easier and faster to train than deeper models while achieving similar performance. The reported results are also invariant to the number of units in the hidden layer.

**Model training.** The parameter  $L$  used in this paper is the average number of action classes for a specific dataset, 6 and 9 for BF and YII, respectively. Empirical experiments showed that using a single hidden layer was easier and faster to train than deeper models while achieving similar performance. The reported results are also invariant to the number of units in the hidden layer. The architecture used to obtain our features is a multi-layer perceptron with as many units as the input feature dimensionality,  $n$ , although this could be changed to obtain the desired output dimensionality. The batch size is equivalent to the downsampling and the number of batches will be the quotient of the number of frames and the batch size. We define the distance hyperparameter as the minimum threshold  $\varepsilon$  that the difference of the last two losses should take. This hyperparameter is set to track early stops with a patience of 2 times. The minimum and maximum training epochs are fixed at 2 and 50, respectively. The initial learning rate depends on each dataset and follows an exponential learning decay rate of 0.3 and a weight decay  $L_2$  of  $10^{-3}$  as the regularisation parameter.

## 4.3. Model study

Figure 3 illustrates how the initial features are modified during training to gradually unveil the representational clusters. On the left, we visualize the affinity matrix computed from the feature matrix on the original feature space. In the middle, we visualize the affinity matrix computed at different epochs during training and finally, on the right, our learned representations at the end of the training. Finally, the clusters, that were completely hidden on the initial affinity matrix, become more and more visible along the diagonal. The same label appears at different time intervals in the off-diagonal clusters, as reflected by the ground truth.

**Ablation study.** In Table 2, we show the importance of modelling both temporal and semantic similarities, by using both  $f_t$  and  $f_s$  (being  $f_{ts}$  when both are marked) and the effectiveness of our network as opposed to adding one more hidden layer or using another function, Radial Basis Function Neural Networks (RBFNN). We use it to compare with our problem because RBFNN are particularly good at modelling non-linear decision boundaries, making RBF networks well suited to our problems [3]. In Table 3, we can see that by representing a frame through a PDF instead of a simpler feature vector, we can significantly improve the accuracy of action segmentation. Without this representation, the approach may not accurately capture the similarity between points in  $X$ , leading to segmentation errors. In addition, the average calculation time tells us that this improvement is achieved with no additional cost.

Breakfast Action Dataset				
Baselines	MoF	IoU	F1	T
Equal Split*	34.8	21.9	-	✗
Spectral*	55.5	44.6	-	✗
Kmeans*	42.7	23.5	-	✗
FINCH* [27]	51.9	28.3	-	✗
Unsupervised				
LSTM+AL [1]	42.9	46.9	-	✓
VTE [30]	52.2	-	-	✓
DGE* [6] (Kmeans)	58.8	47.8	51.6	✗
DGE* [6] (Spectral)	59.5	48.5	51.7	✗
TW-FINCH [26]	62.7	42.3	49.8	✗
ABD [8]	<u>64.0</u>	-	52.3	✗
Ours* (Kmeans)	63.7	<b>53.3</b>	<b>58.0</b>	✗
Ours* (Spectral)	63.2	<u>52.7</u>	<u>57.8</u>	✗
Ours* (FINCH)	<b>65.1</b>	52.1	54.6	✗

Youtube INRIA Instructional Dataset			
Baselines	F1	MoF	T
Equal Split*	27.8	30.2	✗
Spectral*	44.6	55.1	✗
K-means*	29.4	38.5	✗
FINCH* [27]	35.4	44.8	✗
Unsupervised			
LSTM+AL [1]	39.7	-	✓
DGE* [6] (Kmeans)	47.0	42.1	✗
DGE* [6] (Spectral)	48.9	44.8	✗
TW-FINCH [26]	48.2	56.7	✗
ABD [8]	49.2	<b>67.2</b>	✗
Ours* (Kmeans)	<b>55.3</b>	59.7	✗
Ours* (FINCH)	<u>54.7</u>	<u>62.4</u>	✗

Table 1. Action Segmentation results on the BF and YII dataset by applying the Hungarian matching at the video-level. T denotes whether the method has a training stage on target activity/videos. The dash indicates “not reported”. \* denotes results computed by ourselves. The best and second-best results are marked in bold and underlined, respectively.

$f_t$	$f_s$	BF (kmeans)		BF (FINCH)		YII (kmeans)		YII (FINCH)	
		F1	MoF	F1	MoF	F1	MoF	F1	MoF
✗	✓	38.6	44.4	35.3	49.0	46.8	52.8	44.1	53.7
✓	✗	57.7	63.5	54.0	64.6	54.8	59.4	53.5	62.2
✓	✓	<b>58.0</b>	<b>63.7</b>	<b>54.6</b>	<b>65.1</b>	<b>55.3</b>	<b>59.7</b>	<b>54.7</b>	<b>62.4</b>
<b>MLP 2 LAYER</b>		57.2	63.2	52.0	63.4	54.4	59.1	52.9	61.8
<b>RBFNN</b>		20.9	50.2	19.1	49.8	12.9	42.4	10.4	42.6

Table 2. Ablation studies the BF and YII datasets, showing the importance of modelling both temporal and semantic information, and the effectiveness of the single layer MLP network as opposed to adding one more layer or using RBFNN.

#### 4.4. Experimental results

To obtain the final segmentation from the learned action representations we apply three different clustering algorithms: K-means, Spectral clustering and FINCH [27]. For comparison purposes, we report here the results of existing unsupervised methods that were computed by the proposing authors by applying the Hungarian matching at the video-level. We used the code made publicly available by the authors<sup>2</sup> to compute the performance of DGE on the considered datasets, since this approach, similar to ours, computes a video representation suitable for the task of temporal/action segmentation.

**Breakfast Action:** The left-hand table 1 reports the resulting metrics for the BF dataset, obtained with a learning rate 0.051, distance 0.032 and batch size 128. Our method significantly outperforms all other existing approaches. Special attention on F1, which is considerably better in our results, which tells us better quality and less over-segmentation in our method. These results are consistent with all three clustering approaches considered for obtaining the final segmentation of our learned features. We can therefore conclude that TSA outperforms SoTA approaches for the downstream task of action segmentation. Examples of segmentation results on a few videos for this dataset can be seen in Figure 3 (b) and Figures 4 (a)-(c).

<sup>2</sup><https://github.com/mdimiccoli/DGE>

PDF	BF (FINCH)			YII (FINCH)		
	F1	MoF	Time (seconds)	F1	MoF	Time (seconds)
✗	50.5	57.7	32.521	48.4	56.4	11.307
✓	<b>54.6</b>	<b>65.1</b>	32.935	<b>54.7</b>	<b>62.4</b>	11.004

Table 3. Comparative results obtained by using the triplet loss with and without a PDF of similarities, and the average running time for a video in the BF (left) and YII (right) datasets.

**Youtube INRIA Instructional:** The right-hand table 1 reports the resulting metrics for the YII dataset, obtained with a learning rate 0.403, distance 0.892 and batch size 12. This dataset is particularly challenging because of the nature of the annotations, where most of the frames in each video are labelled as background frames. To enable direct comparison, we follow the same procedure used in previous work [8, 26, 28] and report results by removing the ratio ( $\tau = 75\%$ ) of the background frames from the video sequence and then report the performance. To capture the temporal dependencies in the time window, we compute the temporal similarity matrix before subtracting a ratio of the background frames. Our method improves the best *F1* metrics from the literature with a large margin, which indicates the quality of the segmentation in our method on both datasets, as the MoF does not reflect the quality, especially when the whole sequence is dominated by some very long segments. A segmentation output sample of our method for this dataset is plotted in Figures 4 (d)-(f). Also for this dataset, our results are consistent with all two clustering approaches.

It is important to highlight the cases in Figures 4 (a), (b) and (d) where the effectiveness of our approach can be clearly observed. In these examples, the MoF values are comparable in both clustering methods, with the TW-FINCH method exhibiting slightly higher MoF values. However, it is noteworthy that our F1 is significantly better which shows a better quality of segmentation.

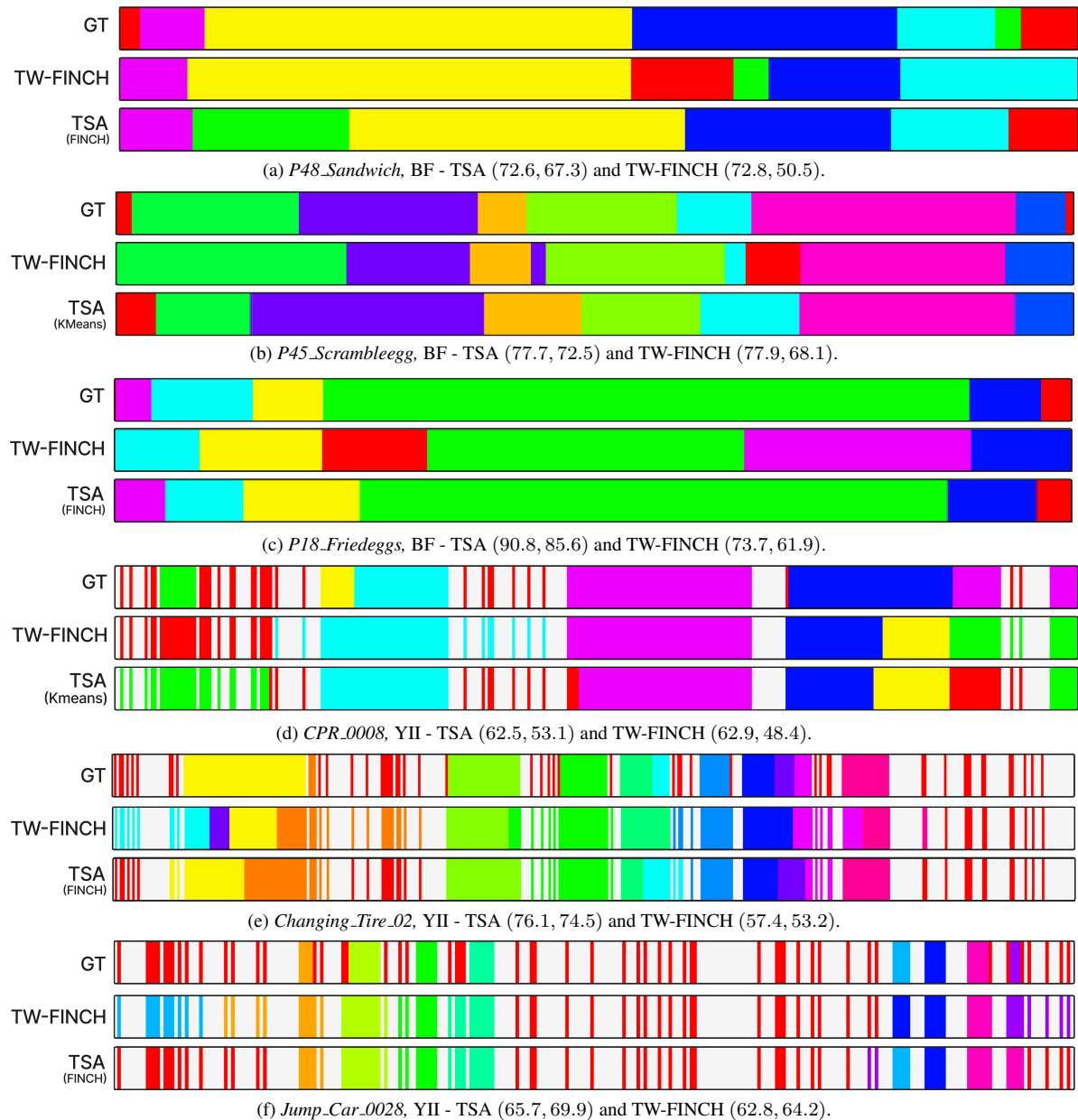


Figure 4. Segmentation output comparisons on two sample videos from BF and YII. Each caption shows the name of the video and the results  $(x, y)$  which are (MoF, F1) for each example.

## 5. Conclusions

This paper introduced a novel fully unsupervised approach for learning action representations in complex activity videos that solely operates on a single unlabelled input video. Our method exploits the temporal proximity and the semantic similarity in the initial feature space to discover the representational clustering grounding action segmentation. Our key contributions are a shallow architecture and a triplet-based loss with a triplet-based selection mechanism based on similarity distribution probabilities to model tem-

poral smoothness and semantic similarity within and across actions. Experimental results on the BF and the YII datasets demonstrated that the learned representations, followed by a generic clustering algorithm, achieve SoTA performance. Furthermore, it has the advantage of not requiring human annotations, is easy to train and does not present domain adaptation issues. Future work will consider how to jointly learn the action clusters and the representation as well as how to build on representations learn at the video-level to match videos at the activity-level.

## Acknowledgments

This work was supported by the project PID2019-110977GA-I00 funded by MCIN/ AEI/ 10.13039/501100011033 and by "ESF Investing in your future".

## References

- [1] Sathyanarayanan N. Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 5, 6
- [2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4575–4583, 2016. 1, 2, 5
- [3] Alireza Asvadi, M Karami, and Yasser Baleghi. Efficient object tracking using optimized k-means segmentation and radial basis function neural networks. volume 4, pages 29–39, 2011. 5
- [4] Catarina Dias and Mariella Dimiccoli. Learning event representations by encoding the temporal context. In *Workshops Computer Vision (ECCV)*, volume 11131, pages 587–596, 2018. 3
- [5] Mariella Dimiccoli and Herwig Wendt. Enhancing temporal segmentation by nonlocal self-similarity. *International Conference on Image Processing (ICIP)*, pages 3681–3685, 2019. 3
- [6] Mariella Dimiccoli and Herwig Wendt. Learning event representations for temporal segmentation of image sequences by dynamic graph embedding. *IEEE Transactions on Image Processing*, 30:1476–1486, 2021. 3, 6
- [7] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6508–6516, 2018. 1, 2
- [8] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3313–3322, 2022. 1, 2, 3, 5, 6
- [9] Yazan Abu Farha and Juergen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2
- [10] Mohsen Fayyaz and Jürgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 498–507, 2020. 1, 2
- [11] De-An Huang, Li Fei-Fei, and Juan Carlos Nieves. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference Computer Vision (ECCV)*, pages 137–153, 2016. 1, 2
- [12] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 780–787, 2014. 5
- [13] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *Winter Applications for Computer Vision(WACV)*, 2016. 1, 2
- [14] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 1, 2
- [15] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12066–12074, 2019. 1, 2, 5
- [16] Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M. Zeeshan Zia, and Quoc-Huy Tran. Unsupervised action segmentation by joint representation learning and online clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20174–20185, 2022. 1, 3
- [17] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2
- [18] Colin Lea, Austin Reiter, René Vidal, and Gregory D. Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European Conference Computer Vision (ECCV)*, pages 36–52, 2016. 1, 2
- [19] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *International Conference on Computer Vision (ICCV)*, pages 6242–6250, 2019. 1, 2
- [20] Jun Li and Sinisa Todorovic. Action shuffle alternating learning for unsupervised action segmentation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12623–12631, 2021. 1, 2
- [21] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2
- [22] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *International Conference on Computer Vision (ICCV)*, pages 8085–8095, 2021. 1, 2
- [23] Yan Bin Ng and Basura Fernando. Weakly supervised action segmentation with effective use of attention and self-attention. *Computer Vision and Image Understanding*, 213:103298, 2021. 1, 2
- [24] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1273–1282, 2017. 1, 2
- [25] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7386–7395, 2018. 1, 2
- [26] M. Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhamen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Conference on Computer Vision and Pattern*



- Recognition (CVPR)*, pages 11225–11234, 2021. [1](#), [2](#), [3](#), [5](#), [6](#)
- [27] M. Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2019. [3](#), [6](#)
- [28] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8368–8376, 2018. [1](#), [2](#), [5](#), [6](#)
- [29] Dipika Singhania, Rahul Rahaman, and Angela Yao. Iterative contrast-classify for semi-supervised temporal action segmentation. In *Conference on Artificial Intelligence (AAAI)*, 2022. [1](#), [2](#)
- [30] Rosaura G. VidalMata, Walter J. Scheirer, Anna Kukleva, David D. Cox, and Hilde Kuehne. Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 1237–1246, 2021. [1](#), [2](#), [5](#), [6](#)
- [31] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *International Conference on Computer Vision (ICCV)*, pages 3551–3558. IEEE, 2013. [5](#)
- [32] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *Winter Conference on Applications of Computer Vision (WACV)*, 2022. [1](#), [2](#)
- [33] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, 2015. [1](#), [2](#)