Random Walks for Temporal Action Segmentation with Timestamp Supervision

Roy Hirsch Regev Cohen Tomer Golany Daniel Freedman Ehud Rivlin

Verily AI, Israel
royhirsch@verily.com

Abstract

Temporal action segmentation relates to high-level video understanding, commonly formulated as frame-wise classification of untrimmed videos into predefined actions. Fully-supervised deep-learning approaches require dense video annotations which are time and money consuming. Furthermore, the temporal boundaries between consecutive actions typically are not well-defined, leading to inherent ambiguity and inter-rater disagreement. A promising approach to remedy these limitations is timestamp supervision, requiring only one labeled frame per action instance in a training video. In this work, we reformulate the task of temporal segmentation as a graph segmentation problem with weakly-labeled vertices. We introduce an efficient segmentation method based on random walks on graphs, obtained by solving a sparse system of linear equations. Furthermore, the proposed technique can be employed in any one or combination of the following forms: (1) as a standalone solution for generating dense pseudo-labels from timestamps; (2) as a training loss; (3) as a smoothing mechanism given intermediate predictions. Extensive experiments with three datasets (50Salads, Breakfast, GTEA) show that our method competes with state-of-the-art, and allows the identification of regions of uncertainty around action boundaries.

1. Introduction

Video understanding covers a wide range of problems concerned with automatically extracting information from videos. In particular, it includes the task of temporal action segmentation which plays an important role in various applications such as autonomous driving [3], robotics [25, 42] and healthcare [7–9, 16, 19–21, 30], amongst others.

In its most basic form, temporal action segmentation aims at classifying each frame of an untrimmed video into one of a set of predefined actions. In contrast to standard image classification, the prediction at a specific time-point within the video also relies on other time-points to capture and utilize temporal dependencies between frames. In recent years, fully-supervised techniques have achieved unprecedented performance in action segmentation. Yet, this level of supervision suffers from a critical drawback as it requires frame-wise annotations which are extremely expensive and time consuming. In addition, the transitions between consecutive actions are not well-defined in time, thus, creating inherently-ambiguous time-intervals which in turn lead to inter-rater disagreement and unreliable annotations. To deal with this, unsupervised methods have been proposed which require little or no information, thus, removing the limitation described above. Unfortunately, their performance is dramatically inferior to that of fully-supervised techniques, rendering them impractical.

As a compromise between the two opposing approaches discussed above, paradigms of weak-supervision have been studied in recent years. Among these paradigms, a promising approach is timestamp supervision [34, 37] which has shown clear potential in reaching the performance of fully-supervised algorithms using only a fraction of the annotated data. The settings of timestamp supervision entails annotating only a single frame of every action instance within a training video. Typically, timestamps are selected as representative frames of their related actions, thus, they are situated outside of ambiguous intervals. Thus, timestamp supervision significantly facilitates the annotation process [36] while providing strong temporal cues for action segmentation. The most prominent approach utilizes timestamps to generate dense pseudo-labels and then trains a temporal segmentation model in a supervised fashion.

In this work, we approach the task of temporal action segmentation with timestamp supervision by representing it as a graph segmentation problem. Here any given video induces a sparse graph whose vertices are the video frames, labeled according to the timestamps provided. An edge weight between two arbitrary frames is determined by their local proximity in time and their feature similarity. Inspired by [17], we then perform action segmentation via random walks on graphs, described as follows. Starting at an unlabeled vertex/frame, we determine the probability we will first reach a certain timestamped vertex, for all timestamps.

Then, we assign a label to the unlabeled vertex according to the timestamp of the greatest probability. Computing the above for all vertices jointly translates to solving a sparse system of linear equations, which can be done efficiently using various well-established techniques. The derived method for action segmentation can be utilized in any one or a combination of the following forms:

1. Given per-frame features (e.g. extracted from a pre-trained model) and timestamps, the random walk can be used as a standalone solution for producing dense predictions, i.e. pseudo-labels.
2. The random walk formulation induces an objective that can be utilized as an auxiliary loss for training action segmentation models with timestamp supervision.
3. Given per-frame features and predictions, our random walk solution can be used as a refinement mechanism to smooth the predictions. Note that the features and the predictions may originate from different sources.

As noted above, we may apply any of the three approaches individually. However, as we aim to provide a complete and consistent solution, we focus here on producing a unified action segmentation technique which applies all of the aforementioned modes of random walks. We evaluate our method on three action segmentation benchmarks: 50salads [46], Breakfast [15] and GTEA [26]. Our results show that our random walk action segmentation method competes with state-of-the-art techniques in timestamp supervision. In summary, our main contributions are as follows:

- We introduce a random walk segmentation (RWS) method which views action segmentation with timestamp supervision as a graph segmentation problem.
- The proposed method can be used as either complete or complementary action segmentation solution. This includes using our random walk as a training objective, a smoothing mechanism or a pseudo-label generator.
- Extensive experiments demonstrate our performance is comparable to or better than state-of-the-art.
- A novel analysis of the uncertainty prediction of the temporal action segments. We show that in contrast to competing methods, our technique produces smooth action transitions, allowing to better identify uncertainty regions near action boundaries.

2. Related Work

Fully Supervised Action Segmentation. Traditional supervised approaches for action segmentation involve a two-step procedure for extracting per-frame features and fusing them together over time. The temporal integration can be performed using either hidden markov models (HMM) [27, 28, 32] or using a parametric recurrent neural network (RNN) [39, 45]. More recent approaches use the parameters' efficient dilated convolution operation in order to increase the receptive field of the model and allow longer patterns to be captured. Two paradigms for using Temporal Convolution Networks (TCNs) have been proposed: an Encoder-Decoder architecture [10, 31], and a multi-stage architecture (MS-TCN) [12, 35]. The latter exhibits improved results since it comprises multiple cascaded stages, each containing a stack of dilated convolution layers, which gradually refine the predictions. A recent approach, ASFormer [50] utilizes the transformer architecture [47] for action segmentation. It consists of an encoder and several decoders for refinement and exhibits comparable results to MS-TCN. UVAST [2], is a recent work that also utilizes the transformer architecture, and aims to directly predict the actions' sequence (*transcript*, see below) and their duration from the video frames. It improves the segments' order prediction, but is less accurate in predicting the boundaries' locations. In spite of their impressive results, these methods require a large amount of carefully annotated videos, which is expensive and time-consuming to collect.

Unsupervised Action segmentation. The challenge of predicting temporal actions without using annotated data has been addressed by several methods. Some use pre-training tasks for learning frame-wise features. The refined representations are then clustered to form segments [1,29,49]. Another line of research offers temporally-aware clustering methods using pre-trained image encoders for extracting the frames representations [11, 43]. TW-FINCH [43] is a recent approach for temporally weighted hierarchical clustering. It uses temporal and spatial features to represent a video as a 1-nearest neighbor graph that is iteratively partitioned. In comparison to fully-supervised or weakly-supervised approaches, the above methods suffer from significantly degraded performance.

Weakly Supervised Action Segmentation. The use of weakly-supervised approaches has been proposed as a solution to balance reasonable performance and data efficiency. Many works rely on *transcripts* supervision, a per-video list of ordered actions. They try to align the frames and transcripts using different techniques such as Viterbi [41] and Dynamic Time Warping [5, 6]. Other approaches rely on a set of action labels (*action sets*), without knowing their temporal location, order and frequency [13, 33, 40]. Some use even weaker supervision in the form of complementary textual data such as narrations or subtitles [14, 44]. Typically, these approaches are designed to address a specific type of weak supervision and result in inferior performances.

Timestamp-Based Action Segmentation. Li *et al.* [34] were the first to introduce the use of timestamp supervision for temporal action segmentation as a form of weak supervision. Their proposed method, ABE, presents a greedy al-

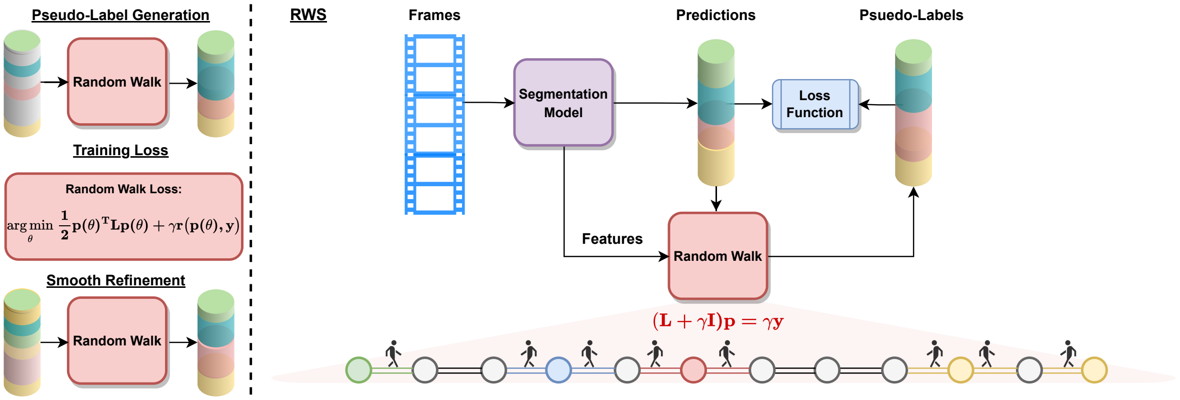
Pseudo-Label Generation
Random Walk
Training Loss
Random Walk Loss:
$\arg\min_{\theta} \frac{1}{2}\mathbf{p}(\theta)^{\mathbf{T}}\mathbf{L}\mathbf{p}(\theta) + \gamma\mathbf{r}(\mathbf{p}(\theta), \mathbf{y})$
Smooth Refinement
Random Walk
RWS
Frames
Segmentation Model
Predictions
Loss Function
Psuedo-Labels
Features
Random Walk
$(\mathbf{L} + \gamma\mathbf{I})\mathbf{p} = \gamma\mathbf{y}$

Figure 1. An overview of the proposed segmentation framework. We develop a random walk approach for temporal action segmentation which accepts per-frame features and weak labels of an input video, models it as a graph and propagates the labels between vertices/frames. The proposed system can be used to generate pseudo-labels, as a training loss or to refine given predictions at inference time. All mentioned modes are utilized to create a unified random walk segmentation (RWS) technique for action segmentation with timestamp supervision.

gorithm for estimating the action boundaries based solely on timestamps. Using the dense pseudo-labels, they train a parametric action segmentation model (MS-TCN). Their method achieves promising results, but its sequential nature makes it sub-optimal. The graph-based method we propose better models the long-range dependencies between frames. Bermann *et al.* [2] present an iterative clustering method for action boundaries estimation and integrate it with their proposed model, UVAST. Our proposed method for pseudo-labels prediction is more efficient and deals more carefully with boundary prediction. Additionally, we present an improved mechanism for temporal smoothing based on the idea of encoding the frames into a graph. Khan *et al.* [22] also propose modelling the video as a graph. However, they alternate between optimizing two functions, a graph convolutional network for frame-wise label generation and an action segmentation model, leading to inefficiency. Rahaman *et al.* [38] recently proposed an iterative expectation-maximization method[1] for action segmentation, which trains a model with a weighted cross-entropy loss. The weights are determined adaptively to capture the uncertainty of unlabeled frames, similar to our weighted smoothing function.

3. Methodology

3.1. Problem Definition

Let an input video $V = [\mathbf{x}_1, ..., \mathbf{x}_N]$ be a sequence of N frames where $\mathbf{x}_i$ denotes the frame at time i. The video consists of $S \ll N$ consecutive temporal segments, each corresponds to one of C predefined action labels $A = [C] \triangleq \{1, ..., C\}$. In timestamp supervision, we are provided with S ground-truth labels $Y = \{y_j\}_{j=1}^{S}$ ($|Y| = S$) at time-points $\{t_j\}_{j=1}^{S}$, one for each temporal segment. Given the timestamps, action segmentation aims at producing dense predictions[2] $[\hat{a}_1, ..., \hat{a}_N]$, i.e., classifying each frame x_i to an action label $\hat{a}_i \in A$.

3.2. Random Walks for Action Segmentation

Our work builds on the random walk algorithm for image segmentation [17], which we adapt the for action segmentation. We start by representing the input video as an undirected weighted graph $G = (\mathbf{V}, \mathbf{E}, \mathbf{W})$, with each vertex $\mathbf{v}_i \in \mathbf{V}$ ($|\mathbf{V}| = N$) associated with a frame $\mathbf{x}_i$ and the edges $\mathbf{E} = \{e_{i,j}\}$ connect successive frames. The edge weights encode vertex similarity as follows

$$w_{i,j} \triangleq \exp\left(-\beta||\mathbf{f}_i - \mathbf{f}_j||^2\right) \tag{1}$$

where $\mathbf{f}_i$ are the features of $\mathbf{x}_i$ and $\beta > 0$ is an hyperparameter. The timestamps induce a partition $\mathbf{V} = \mathbf{V}_Y \cup \mathbf{V}_U$ ($\mathbf{V}_Y \cap \mathbf{V}_U = \emptyset$) where $\mathbf{V}_Y$ ($|\mathbf{V}_Y| = S$) is the set of seed vertices corresponding to timestamped frames. The random walk algorithm aims to compute for each unlabeled vertex $\mathbf{v}_i \in \mathbf{V}_U$ the probability p_a^i that a random walker starting at $\mathbf{v}_i$ first reaches a vertex timestamped with action a. Finally, the vertex is assigned to the label of the highest probability. As shown in [18], the probabilities can be computed by minimizing for all actions $a \in A$ the following objective

$$Q(\mathbf{p}_a) = \frac{1}{2}\mathbf{p}_a^T\mathbf{L}\mathbf{p}_a \quad \text{s.t.} \quad p_a^{t_j} = \mathbb{1}[a = y_j] \;\; \forall \mathbf{v}_j \in \mathbf{V}_Y, \tag{2}$$

where we define $\mathbf{p}_a \triangleq [p_a^1, ..., p_a^N]$. Here $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$ is the graph Laplacian with $\mathbf{D}$ being a diagonal matrix with entries $\mathbf{D}_{ii} = \sum_j w_{i,j}$. Thus, the probabilities are obtained by solving a sparse, positive-definite, system of lin-

[1] To our understanding, this model was trained on a different set of timestamps that the the one we experiment with, adhering to [34].

[2] In practice, the output may not be per-frame predictions, as in [2]. Yet, any form of valid output can be translated to per-frame predictions.

ear equations instead of performing a random walk simulation. Moreover, since the probabilities at any vertex must sum to unity, only $C-1$ linear systems must be solved.

3.3. Regularized Random Walks

While the classic random walk algorithm provides a rigorous and efficient approach to action segmentation by reducing the problem to solving systems of linear equations, it has two important limitations. First, it only considers edges between neighboring frames, thereby neglecting important temporal information. Second, the algorithm treats timestamps as hard constraints, precluding label modification of the corresponding vertices. To alleviate these limitations, we perform the following modifications. We extend the set of edges to $\mathbf{E} \triangleq \{(i,j) \mid |i-j| \leq m\}$ where $m > 0$ is an hyperparameter. To further improve the temporal context, we adjust the weights as follows

$$\tilde{w}_{i,j} \triangleq \begin{cases} \texttt{Agg}(w_{i,j-K}, ..., w_{i,j}, ..., w_{i,j+K}), & (i,j) \in \mathbf{E}, \\ 0, & \textit{otherwise}. \end{cases} \tag{3}$$

where $K > 0$ controls the context window size, and $\texttt{Agg}$ is an aggregation function set as either *min*, *max*, *mean* or *none*. The latter implies simply setting $w_{i,j} \equiv \tilde{w}_{i,j}$. Next, we introduce regularization into the objective function

$$\hat{\mathbf{p}}_a = \arg\min_{\mathbf{p}_a} \frac{1}{2}\mathbf{p}_a^T \mathbf{L} \mathbf{p}_a + \gamma r(\mathbf{p}_a, \mathbf{y}_a). \tag{4}$$

Here $\gamma > 0$ and $r(\cdot,\cdot)$ measures similarity to a supervision vector $\mathbf{y}_a$, where in our setup it is a binary vector whose non-zero entries correspond to timestamps with label a. For simplicity we set $r(\mathbf{p}_a, \mathbf{y}_a) \triangleq ||\mathbf{H}_a(\mathbf{p}_a - \mathbf{y}_a)||^2$ where $\mathbf{H}_a \triangleq \texttt{diag}(\mathbf{y}_a)$. Consequently, we compute the probabilities by solving the following sparse system of linear equations

$$(\mathbf{L} + \gamma \mathbf{H}_a)\mathbf{p}_a = \gamma \mathbf{y}_a. \tag{5}$$

Finally, we set our predictions as $\hat{a}_i = \arg\max_a \ p_a^i$.

The proposed regularized random walk algorithm retains all the computational efficiency benefits of the original method, while offering flexibility in determining the temporal context size and in adhering to the timestamps.

3.4. Modes of Operation

Our method engenders multiple modes of operation:

Pseudo-Label Generation. By solving (5) we can use timestamps to generate dense pseudo-labels. This procedure can be seen as a standalone solution which requires only per-frame features $F = \{\mathbf{f}_i\}$ to build the Laplacian and performs no training. Alternatively, we can solve (5) repeatedly at training time to update the pseudo-labels with any update of the model features.

Smooth Refinement. At test time, given per-frame features $F = \{\mathbf{f}_i\}$ and the model predictions $\{\mathbf{y}_a\} \triangleq \{\tilde{\mathbf{p}}_a\}$, we first build the Laplacian $\mathbf{L}$ using the features and then utilize (5) as a refinement mechanism, adjustable via γ, to produce smooth results. Notice that the features and predictions may originate from different models, offering flexibility.

As Training Losses. Let $\mathbf{p}_a = \mathbf{p}_a(\theta)$ be the output of a segmentation network $\mathcal{M}$ with parameters θ. Revisiting (4), we define a Laplacian-based smoothing loss function

$$\mathcal{L}_{lap} \triangleq \sum_{a\in A} \mathbf{p}_a^T(\theta)\mathbf{L}\mathbf{p}_a(\theta) = \sum_{a\in A}\sum_{(i,j)} w_{i,j}|p_a^i - p_a^j|^2, \tag{6}$$

where $\mathbf{L}$ can depend on the model parameters θ or on external features. In practice, we use a truncated version

$$\mathcal{L}_{T-Lap} \triangleq \sum_{a\in A}\sum_{(i,j)} w_{i,j}\Delta_{i,j}^2, \tag{7}$$

where for a given $\tau > 0$ we define

$$\Delta_{i,j} \triangleq \begin{cases} |p_a^i - p_a^j|, & |p_a^i - p_a^j| \leq \tau, \\ \tau, & \text{otherwise}. \end{cases} \tag{8}$$

The second term in (4) can be substituted with any supervised loss function $\mathcal{L}_{sup}$. Thus, we train $\mathcal{M}$ using the following random walk loss: $\mathcal{L}_{RW} = \mathcal{L}_{sup} + \mathcal{L}_{T-Lap}$.

3.5. Unified Random Walk Solution

So far, we presented our random walk and its operating modes. Here we join the different modes to create a unified random walk solution for temporal action segmentation with timestamp supervision. Our random walk segmentation (RWS) method consists of the following stages:

1. **Pseudo-Label Generation**. Before training, we use the timestamps and per-frame features, extracted by a pre-trained model, to yield dense pseudo-labels via (5).
2. **As Training Losses**. During training, we set our action segmentation model as an MS-TCN [48] and train it using the following loss

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{T-Lap} + \delta\mathcal{L}_{conf}, \tag{9}$$

where $\mathcal{L}_{cls}$ is a classification cross-entropy loss computed over the generated pseudo-labels, $\mathcal{L}_{conf}$ is the confidence loss [34], given in supplementary, and $\alpha > 0$ and $\delta > 0$ are hyperparameters denoting the contribution of each loss. Following each update step of the model, we use it to extract per-frame features and update our pseudo-labels and Laplacian term $\mathcal{L}_{T-Lap}$.
3. **Smooth Refinement**. At inference, we run our model to obtain per-frame predictions $\{\hat{\mathbf{p}}_a\}$. We then apply our random walk with a pre-trained features to refine our predictions by solving (5) with $\mathbf{y}_a = \hat{\mathbf{p}}_a$ and $\mathbf{H}_a = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix.

4. Experiments

We validate the effectiveness of our proposed method for pseudo-label generation and our unified random walk solution. We also ablate the design choices and optimization process, and provide insights into the unexplored feature of estimating the confidence of action segmentation models.

4.1. Experimental Setting

4.1.1 Datasets

We evaluate the performance of the proposed model over the next challenging datasets - (i) **Breakfast** [15]: 1712 videos of breakfast preparation actions, each frame is annotated with one of 58 action classes. Each video contains 130-9000 frames and 2-25 segments; (ii) **50Salads** [46]: 50 videos of people preparing different salads, each frame is annotated with one of 17 action classes. Each video contains 7000-18000 frames and 15-26 segments; (iii) **GTEA** [26]: 28 videos of actors performing various daily activities, each frame is annotated with one of 11 actions. Each video contains 630-2000 frames and 21-44 segments. For a fair comparison, we use the same random timestamps as in [34]. We follow previous works [12, 35, 50] and perform 4-fold cross-validation on Breakfast and GTEA, 5-fold cross-validation on 50Salads and report the mean results.

4.1.2 Metrics

To be consistent with previous works, we report the following metrics - (i) **Frame-wise accuracy (Acc)**: Mean classification accuracy, computed over all of the frames within a video. This metric assigns larger weights to longer segments and focuses more on boundary locations; (ii) **Segmental edit score (Edit)** [32]: Measures the alignment of the predicted transcript (ordered sequence of actions) to the ground true transcript, without considering the segment duration; (iii) **Segmental F1 (F1@10,25,50)** [31]: This metric compares the intersection over union (IoU) of each segment with respect to the corresponding ground truth. A segment is considered a true positive if its IoU in relation to the ground truth exceeds a selected overlapping threshold. The metric is measured at a few thresholds of 10%, 25% and 50%. This score penalizes over-segmentation and is less sensitive to the exact location of the boundaries. The segmental F1 score provides a more comprehensive measure of the segmentation quality [2, 31], as the action boundaries are ambiguous by nature. Therefore, we put emphasis on this measure throughout our empirical analysis.

4.1.3 Competing Methods

We compare our approach, RWS , with current state-of-the-art methods for timestamp-based temporal segmentation:

Action Boundary Estimation (ABE) [34]. A fundamental work that introduced and applied the use of timestamps as weak supervision for temporal segmentation. It introduces a heuristic for predicting per-frame actions from timestamps and uses it for creating dense supervision for the temporal segmentation model during training. The presented heuristic is based on the fact that there exists a single boundary between every pair of consecutive timestamps. The authors propose a greedy algorithm for partitioning the segment between each pair of timestamps. The method is repeated twice, scanning the frames from left to right and vice versa and averaging the results to estimate the boundaries.

Unified Video Action Segmentation model via Transformers (UVAST) [2]. A recent method for video action segmentation via sequence to sequence translation. The proposed model is a complex system of transformer encoder and two decoders. —First, the encoder is applied to raw frame representations and produces refined dense per-frame features. Then, the first decoder predicts the transcript based on the frame features and the second decoder predicts the segment duration based on the transcript and the frame features. The authors also address the setup of timestamp supervision and derive a method for generating dense labels from timestamps based on k-medoids clustering.

Graph Convolution Network (GCN) [22]. A recent method that models the inter-frame relations as a graph. In contrast to our method, they propose learning a second parametric model, a GCN [24], for predicting the dense pseudo labels. An optimization process alternates between optimizing the GCN and the temporal segmentation model. [3]

4.1.4 Implementation Details

For a fair comparison, we follow [22, 34] and adopt the multi-stage temporal convolutional (MS-TCN) model presented in [48]. This model is composed of 4 stages, each of which contains 10 blocks of dilated convolutions with a hidden dimension of 64. The first stage of the model consist of two parallel convolutions with different kernel sizes of 3 and 5. The resultant two outputs are summed and passed to the following stages. We train the model for a maximum of 60 epochs, monitor the accuracy of the train timestamp predictions, and end training when the accuracy reaches a saturation point. In accordance with previous work [34], the model is trained only for classifying the annotated timestamps during the first 30 epochs. Next, pseudo-labels are generated based on the learned representation derived from the temporal model. We use Adam [23] optimizer and set the learning rate to 0.0005 and the batch size to 8. Video frames are sampled at a rate of 1 fps, and frame representations are extracted from an I3D model [34]

[3] In contrast to ABE and UVAST, the GCN evaluation setup reports average results over three randomly initialized runs.

	Breakfast		*50Salads*		*GTEA*	
	F1@{10,25,50}	Acc	F1@{10,25,50}	Acc	F1@{10,25,50}	Acc
ABE	96.0 87.3 67.6	72.6	99.4 95.0 76.8	79.9	96.6, 86.3, 65.5	75.5
UVAST	95.5 87.5 **70.0**	**76.9**	97.5 90.4 75.6	**81.3**	**99.8 97.7 83.0**	75.3
RWS	**96.5 88.9** 69.5	76.1	**99.7 97.5 81.0**	80.6	96.7 89.4 71.4	**78.6**

Table 1. Comparing methods for generating dense pseudo-labels from timestamps. Best results are bolded while second best are underlined.

pre-trained over Kinetics-400 dataset [4]. Our method is agnostic to the chosen model and to the frame representations. Code for replicating our experiments is available at https://github.com/RoyHirsch/RWS.

4.2. Results

4.2.1 Pseudo-Label Generation

As a first application, we provide a standalone none-parametric solution for generating dense pseudo-labels from timestamps. We use the frame features extracted from a pre-trained I3D model [4] and use the same timestamps as in [34]. Table 1 compares our method to the two clustering methods presented in ABE [34] and in UVAST [2] (since GCN [22] does not generate pseudo-labels, this method is not relevant for the compression). We compute ABE results using their published code, UVAST resuls are obtained from [2]. We report the mean metrics for all the data folds. Our method improves the segmental F1 by an average of $2.3/0.7$ points in compare to the baselines over 50Salads / Breakfast datasets at the expense of marginal decrease in accuracy. The opposite behavior is shown for GTEA.

4.2.2 Unified Random Walk

We evaluate the performance of our unified random walk segmentation method. Similar to the setup presented in [34], we only use timestamp supervision during training, learn a parametric action segmentation model and evaluate it over videos without any additional annotation. Table 2 compares the performance of RWS to the state-of-the-art timestamp-based methods. We also compare to a few recent fully-supervised baselines: MS-TCN [34], a fully supervised version of our model, MS-TCN++ [35], AS-Former [50] and to a fully supervised UVAST [2].

Due to the fact that there is no single timestamp-based model that outperforms all of the metrics, we provide a ranking of the tested measurements. In the most important metric, segmental F1, our method outperforms its competitors. Furthermore, it achieves the second best performance in terms of segmental edit distance, without utilizing a dedicated transcript prediction component (as used by UVAST). In terms of per-frame accuracy, ABE outperforms all the tested baselines. As stated before and in [2], the exact locations of action boundaries are ambiguous by nature. Hence, The F1 and edit scores are more prominent and effective metrics, thus, are more relevant to real-world applications.

Figure 2 presents the segmentation results for two sampled videos from Breakfast and 50Salads datasets. We compare the performance of our method to ABE by re-training the models with the code provided by [34]. We also compare to UVAST using the trained checkpoints published in [2]. Using different colors to code different actions, the bar plot represents the predicted temporal segments. This illustrates that our method is less prone to over-segmentation and yields less false-positive action segments.

4.3. Ablation Studies

4.3.1 Impact of the Random Walk Parameters

Our random walk temporal segmentation method introduces additional hyperparameters to tune. In this section we explore and quantify the influence of the various parameters over the standalone operative mode of our method. Figure 3 summarizes an ablation study and reports accuracy and F1@10 for different configurations over the three datasets (additional details regarding the experimental setup are provided in supplementary materials). First, we study the influence of the weight aggregation method. As can be seen, any type of aggregation is beneficial, yet *min* aggregation outperforms others by a small margin. This aggregation is particularly useful since it highlights action boundaries where the correlation between frame features is low.

The parameter β, which controls the sharpness of the similarity scores has a more prominent effect. As seen from the results, accuracy improves with sharpness until it until it saturates at a value of 30. Increasing γ, the temporal prior weight, reduces prediction accuracy as the algorithm prioritizes precisely classifying sparse timestamps over correct label propagation. Increasing the *number of neighbors* to the limit of 15 (30 neighbors per frame in total) modestly improves accuracy, but further increases degrade performance as graph weights become less sensitive to local changes. This result is consistent with previous works [22].

4.3.2 Impact of the Smoothing Loss Function

Farha *et al.* [12] have been the first to introduce the use of the truncated mean squared error (T-MSE) over the frame-wise log probabilities to enforce temporal consistency, and

	Breakfast			*50Salads*			*GTEA*			*Rank*		
	F1@10,25,50	Edit	Acc	F1@10,25,50	Edit	Acc	F1@10,25,50	Edit	Acc	F1	Edit	Acc
MS-TCN [34]	70.8 67.7 58.6	63.8	77.8	69.9 64.2 51.5	69.4	68.0	85.1 82.7 69.6	79.6	76.1	-	-	-
MS-TCN++ [35]	64.1 58.6 45.9	65.6	67.6	80.7 78.5 70.1	74.3	83.7	88.8 85.7 76.0	83.5	80.1	-	-	-
ASFormer [50]	76.0 70.6 57.4	75.0	73.5	85.1 83.4 76.0	79.6	85.6	90.1 88.8 79.2	84.6	79.7	-	-	-
UVAST [2]	76.7 70.0 56.6	77.2	68.2	86.2 81.2 70.4	83.9	79.5	77.1 69.7 54.2	90.5	62.2	-	-	-
UVAST [2]	**72.0** 64.1 **48.6**	**74.3**	60.2	75.7 70.6 58.2	**78.4**	67.8	70.8 63.5 49.2	**88.2**	55.3	2	**1**	4
GCN [22]	67.9 61.0 45.3	67.0	61.4	75.1 72.3 61.0	67.6	75.1	**81.5 77.5 60.8**	75.6	66.1	3	3	2
ABE [34]	70.5 63.6 47.4	69.9	**64.1**	73.9 70.9 60.1	66.8	**75.6**	78.9 73.0 55.4	72.3	**66.4**	4	4	**1**
RWS	70.9 **64.7** 44.8	71.1	60.2	**76.7 72.8** 55.5	69.3	70.0	80.9 74.1 56.3	76.2	59.3	**1**	2	3

Table 2. Comparison with state-of-the-art timestamp-based (bottom section) and fully-supervised methods (top section). In the rightmost section, the averaged metrics for each timestamp-based model are ranked.

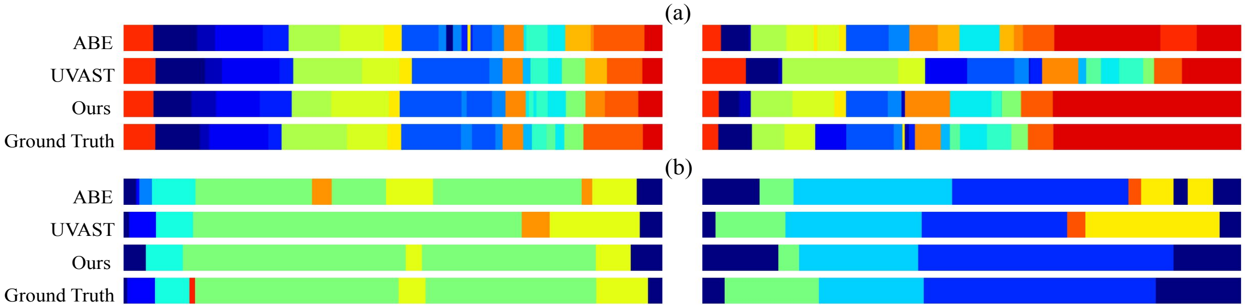

Figure 2. Segmentation results over sample videos from (a) 50Salads and (b) Breakfast datasets.

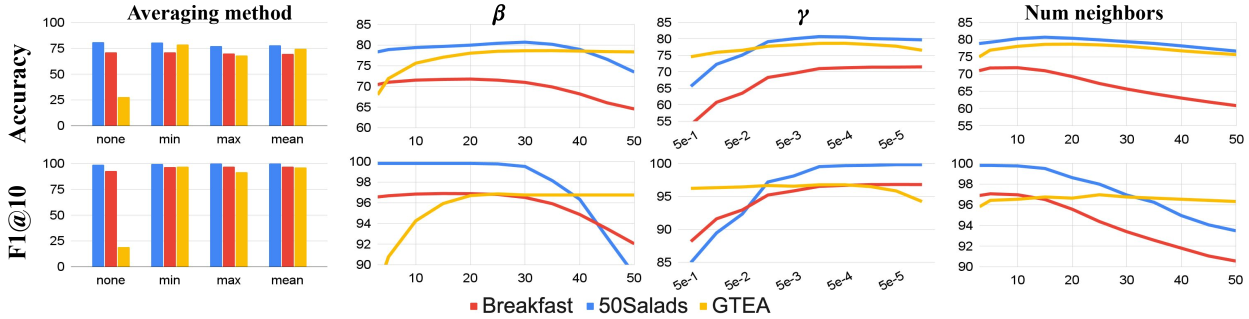

Figure 3. The effect of different parameters over RWS pseudo-dense labels generation results.

thus reduce over-segmentation. Yet, it assigns equal importance to all the temporal locations. Our method introduces a new way to regularize the temporal smoothness based on the graph Laplacian. It enforces a soft temporal consistency that is governed by the graph Laplacian. In other words, as formulated in equation 7, the difference between every two adjacent frames is weighted by their similarity. Furthermore, since edges in our graph are not limited to consecutive frames, using graph Laplacian smoothing is more expressive than using T-MSE, which compares only consecutive frames. In Table 3 we compare between the two smoothing methods over the three tested datasets. As can be seen, the proposed loss achieves improved F1 and accuracy with an improvement of 3.4% and 2.6% respectively. The proposed loss can also be applied to other weakly-supervised or fully-supervised temporal segmentation setups, however, this is outside the scope of our study.

	F1@{10,25,50}	Edit	Acc
	Breakfast		
$\mathcal{L}_{T-MSE}$	**69.4** 62.6 44.5	70.1	57.3
$\mathcal{L}_{T-Lap}$	69.2 **63.1 45.4**	**70.5**	**58.0**
	50salads		
$\mathcal{L}_{T-MSE}$	**75.1** 70.4 50.9	67.3	68.1
$\mathcal{L}_{T-Lap}$	**75.1 71.2 61.1**	**67.8**	**69.6**
	GTEA		
$\mathcal{L}_{T-MSE}$	78.3 71.8 54.9	75.1	56.4
$\mathcal{L}_{T-Lap}$	**80.6 73.4 55.9**	**76.9**	**58.9**

Table 3. Comparing T-Laplacian and T-MSE smoothing.

4.3.3 Impact of the Different Losses

The proposed optimization is composed of three components: a classification loss, a Laplacian smoothing loss, and a confidence loss. Table 4 quantifies the impact of the different terms for Breakfast and 50Salads datasets. Similar to [34], the best results are obtained when using smoothing in conjunction with a confidence term. Using the additional two terms improves the mean segmental F1 by roughly 20%, the edit distance by 12% and the per-frame accuracy by 6.3%. Table 5 quantifies the influence of the Laplacian smoothing over 50Salads dataset. The Laplacian smoothing has less influence over the prediction when its weight is small. Conversely, when its weight is too large, the final prediction is over-smoothed and the performance degrades.

	F1@{10,25,50}	Edit	Acc
	Breakfast		
$\mathcal{L}_{cls}$	63.7 54.7 34.5	67.8	54.9
$\mathcal{L}_{cls} + \mathcal{L}_{T-Lap}$	65.0 54.1 32.6	67.3	56.0
$\mathcal{L}_{cls} + \mathcal{L}_{conf}$	66.1 54.9 33.0	67.8	55.8
$\mathcal{L}_{cls} + \mathcal{L}_{T-Lap} + \mathcal{L}_{conf}$	**69.2 63.1 45.4**	**70.5**	**58.0**
	50Salads		
$\mathcal{L}_{cls}$	68.3 62.6 44.3	60.5	65.0
$\mathcal{L}_{cls} + \mathcal{L}_{T-Lap}$	69.5 64.7 47.0	61.9	66.7
$\mathcal{L}_{cls} + \mathcal{L}_{conf}$	72.7 67.5 49.5	64.8	67.7
$\mathcal{L}_{cls} + \mathcal{L}_{T-Lap} + \mathcal{L}_{conf}$	**75.1 71.1 61.1**	**67.8**	**69.6**

Table 4. The contribution of the different loss terms.

α	F1@{10, 25, 50}	Edit	Acc
0.000	72.7 67.5 49.5	64.8	67.6
0.025	72.7 67.9 49.3	64.0	67.2
0.050	**75.1 71.1 61.1**	**67.8**	**69.6**
0.075	74.1 69.5 51.2	66.3	68.8
0.100	74.0 68.4 50.1	64.2	67.6

Table 5. Impact of the Laplacian smoothing over 50Salads dataset.

4.4. Estimating the Prediction Confidence

A critical aspect of any predictive model is determining the degree of confidence a model has in its predictions. This is a crucial feature in many applications such as autonomous driving [3] and healthcare [9, 16], which is often overlooked in the context of temporal action segmentation. To address this, we compute the entropy of the per-frame probabilities as a measure of the model per-frame confidence, where high entropy implies low confidence and vice versa.

Fig. 4 illustrates the per-frame probabilities and entropy for a sample Breakfast video. Comparing to UVAST is challenging as its per-frame predictions are noisy, and thus refined by two consecutive decoders. Hence, we compare our model to ABE and find that it produces smoother action transitions, which better model natural ambiguity in boundary locations. We quantify this property by measuring the mean accuracy of per-frame predictions at different confidence levels. Table 6 reports accuracy for the kth percentile most confident frames. Considering only high-confidence predictions, RWS outperforms ABE in accuracy by large margins with an average of 6 and 5 percentage points for $k = 60\%/70\%$. Thus, our method assigns low confidence to predictions near action boundaries, identifying ambiguous regions while maintaining high accuracy elsewhere.

	Acc@{90%, 80%, 70%, 60%}	
	RWS	ABE
Breakfast	65.2 80.5 90.4 89.0	65.9 73.2 79.9 76.8
50Salads	73.2 74.3 76.3 78.8	69.5 71.2 73.5 75.6
GTEA	62.6 73.5 82.8 85.4	69.6 74.4 79.2 83.4

Table 6. Comparing per-frame accuracy of the top $k \in \{90\%, 80\%, 70\%, 60\%\}$ most confident frames.

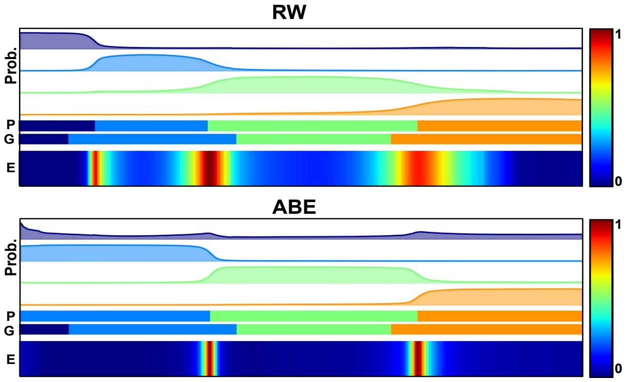

Figure 4. Normalized per-frame probabilities and entropy for a Breakfast dataset video. **P**: predictions, **G**: ground true and **E**: normalized entropy. Our model enforces smoother transitions between action segments, thus, highlights boundaries uncertainty.

5. Conclusions

In this work we presented random walks for temporal action segmentation with timestamp supervision. The proposed random walk can be used to generate dense labels from timestamps, refine given model predictions at inference, and act as an auxiliary training loss. Our RWS method uses all of the aforementioned to provide a unified solution for action segmentation with timestamp supervision. We evaluate our technique on three benchmarks, demonstrating on-par or improved performance with state-of-the-art. Furthermore, our adaptive temporal smoothness mechanism allows identifying ambiguous regions near action boundaries. As future work, the proposed method can be further adapted to accept any form of weak labels, beyond timestamps.

References

[1] Sathyanarayanan N Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1197–1206, 2019. 2

[2] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Jürgen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 52–68. Springer, 2022. 2, 3, 5, 6, 7

[3] Mahdi Biparva, David Fernández-Llorca, Rubén Izquierdo Gonzalo, and John K Tsotsos. Video action recognition for lane-change classification and prediction of surrounding vehicles. *IEEE Transactions on Intelligent Vehicles*, 7(3):569–578, 2022. 1, 8

[4] Joao Carreira and Andrew 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. 6

[5] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019. 2

[6] Xiaobin Chang, Frederick Tung, and Greg Mori. Learning discriminative prototypes with dynamic time warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8395–8404, 2021. 2

[7] Regev Cohen, Yochai Blau, Daniel Freedman, and Ehud Rivlin. It has potential: Gradient-driven denoisers for convergent solutions to inverse problems. *Advances in Neural Information Processing Systems*, 34:18152–18164, 2021. 1

[8] Regev Cohen, Michael Elad, and Peyman Milanfar. Regularization by denoising via fixed-point projection (red-pro). *SIAM Journal on Imaging Sciences*, 14(3):1374–1406, 2021. 1

[9] Tobias Czempiel, Magdalini Paschali, Matthias Keicher, Walter Simson, Hubertus Feussner, Seong Tae Kim, and Nassir Navab. Tecno: Surgical phase recognition with multi-stage temporal convolutional networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23*, pages 343–352. Springer, 2020. 1, 8

[10] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516, 2018. 2

[11] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2022. 2

[12] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019. 2, 5, 6

[13] Mohsen Fayyaz and Jurgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–510, 2020. 2

[14] Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to segment actions from observation and narration. *arXiv preprint arXiv:2005.03684*, 2020. 2

[15] Shang-Hua Gao, Qi Han, Zhong-Yu Li, Pai Peng, Liang Wang, and Ming-Ming Cheng. Global2local: Efficient structure search for video action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16805–16814, 2021. 2, 5

[16] Tomer Golany, Amit Aides, Daniel Freedman, Nadav Rabani, Yun Liu, Ehud Rivlin, Greg S Corrado, Yossi Matias, Wisam Khoury, Hanoch Kashtan, et al. Artificial intelligence for phase recognition in complex laparoscopic cholecystectomy. *Surgical Endoscopy*, pages 1–9, 2022. 1, 8

[17] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006. 1, 3

[18] Leo Grady and Gareth Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *International Workshop on Mathematical Methods in Medical and Biomedical Image Analysis*, pages 230–245. Springer, 2004. 3

[19] Roy Hirsch, Mathilde Caron, Regev Cohen, Amir Livne, Ron Shapiro, Tomer Golany, Roman Goldenberg, Daniel Freedman, and Ehud Rivlin. Self-supervised learning for endoscopic video analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 569–578. Springer, 2023. 1

[20] Roy Hirsch, Regev Cohen, Mathilde Caron, Tomer Golany, Daniel Freedman, and Ehud Rivlin. Weakly-supervised surgical phase recognition. *arXiv preprint arXiv:2310.17209*, 2023. 1

[21] Liran Katzir, Danny Veikherman, Valentin Dashinsky, Roman Goldenberg, Ilan Shimshoni, Nadav Rabani, Regev Cohen, Ori Kelner, Ehud Rivlin, and Daniel Freedman. Estimating withdrawal time in colonoscopies. In *Proceedings of ECCV, MCV workshop*, 2022. 1

[22] Hamza Khan, Sanjay Haresh, Awais Ahmed, Shakeeb Siddiqui, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. Timestamp-supervised action segmentation with graph convolutional networks. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10619–10626. IEEE, 2022. 3, 5, 6, 7

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 5

[25] Volker Krüger, Danica Kragic, Aleš Ude, and Christopher Geib. The meaning of action: A review on action recognition and mapping. *Advanced robotics*, 21(13):1473–1501, 2007. 1

[26] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 2, 5

[27] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016. 2

[28] Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid rnn-hmm approach for weakly supervised temporal action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):765–779, 2018. 2

[29] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019. 2

[30] Gilad Kutiel, Regev Cohen, Michael Elad, Daniel Freedman, and Ehud Rivlin. Conformal prediction masks: Visualizing uncertainty in medical imaging. In *ICLR 2023 Workshop on Trustworthy Machine Learning for Healthcare*, 2023. 1

[31] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017. 2, 5

[32] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal CNNs for fine-grained action segmentation. In *European conference on computer vision*, pages 36–52. Springer, 2016. 2, 5

[33] Jun Li and Sinisa Todorovic. Set-constrained viterbi for set-supervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10820–10829, 2020. 2

[34] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8365–8374, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[35] Y Liu, MM Cheng, SJ Li, YA Farha, and J Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 2, 5, 6, 7

[36] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *European conference on computer vision*, pages 420–437. Springer, 2020. 1

[37] Davide Moltisanti, Sanja Fidler, and Dima Damen. Action recognition from single timestamp supervision in untrimmed videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9915–9924, 2019. 1

[38] Rahul Rahaman, Dipika Singhania, Alexandre Thiery, and Angela Yao. A generalized and robust framework for timestamp supervision in temporal action segmentation. In *European Conference on Computer Vision*, pages 279–296. Springer, 2022. 3

[39] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 754–763, 2017. 2

[40] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5987–5996, 2018. 2

[41] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7386–7395, 2018. 2

[42] Laurel D Riek. Healthcare robotics. *Communications of the ACM*, 60(11):68–78, 2017. 1

[43] Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11225–11234, 2021. 2

[44] Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *Proceedings of the IEEE International conference on Computer Vision*, pages 4480–4488, 2015. 2

[45] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1961–1970, 2016. 2

[46] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738, 2013. 2, 5

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[48] Kanav Vats, Mehrnaz Fani, Pascale Walters, David A Clausi, and John Zelek. Event detection in coarsely annotated sports videos via parallel multi-receptive field 1d convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 882–883, 2020. 4, 5

[49] Rosaura G VidalMata, Walter J Scheirer, Anna Kukleva, David Cox, and Hilde Kuehne. Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1238–1247, 2021. 2

[50] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. *arXiv preprint arXiv:2110.08568*, 2021. 2, 5, 6, 7