Improving Action Segmentation via Graph Based Temporal Reasoning

Yifei Huang, Yusuke Sugano, Yoichi Sato
Institute of Industrial Science, The University of Tokyo
{hyf,sugano,ysato}@iis.u-tokyo.ac.jp

Abstract

Temporal relations among multiple action segments play an important role in action segmentation especially when observations are limited (e.g., actions are occluded by other objects or happen outside a field of view). In this paper, we propose a network module called Graph-based Temporal Reasoning Module (GTRM) that can be built on top of existing action segmentation models to learn the relation of multiple action segments in various time spans. We model the relations by using two Graph Convolution Networks (GCNs) where each node represents an action segment. The two graphs have different edge properties to account for boundary regression and classification tasks, respectively. By applying graph convolution, we can update each node's representation based on its relation with neighboring nodes. The updated representation is then used for improved action segmentation. We evaluate our model on the challenging egocentric datasets namely EGTEA and EPIC-Kitchens, where actions may be partially observed due to the viewpoint restriction. The results show that our proposed GTRM outperforms state-of-the-art action segmentation models by a large margin. We also demonstrate the effectiveness of our model on two third-person video datasets, the 50Salads dataset and the Breakfast dataset.

1. Introduction

Video action segmentation plays a crucial role in various applications such as robotics [31], anomaly detection [7] and human behaviour analysis [56]. The task of action segmentation is to know when and what type of action is observed in a given video. This is done by temporally locating each action segment in the video and classifying the action category of the segment.

The topic of action segmentation has long been studied by the computer vision community. Earlier approaches address this problem by applying temporal classifiers on top of low-level video features, *e.g.* I3D [6] features. They include 1) sliding window approaches [29, 51], which typically have very limited temporal receptive fields; 2) segmental models [36, 46], which have difficulty in capturing long-range action patterns since an action is only conditioned on its previous segment; and 3) recurrent networks [23, 53], which have a limited span of attention [53]. Recently temporal convolutional networks [37] demonstrated a promising capability of capturing long-range dependencies between video frames [14, 16, 35], leading to good results on third-person videos seen from a fixed viewpoint.

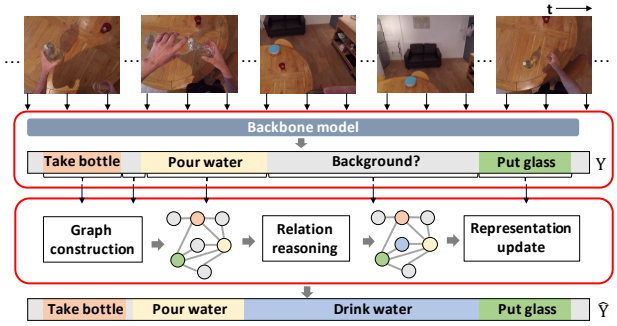

Figure 1. Consider the example video in this figure. The backbone model prunes to detect the segment after *pour water* to be *background* since no action is directly observable from the video. By adding our proposed GTRM on top, we can successfully detect this segment to be *drink water* by learning the temporal relation between the actions. The relation among multiple actions can also help to adjust the segment boundaries.

However, it remains difficult for existing methods to work well when only limited observations are available (*e.g.* due to occlusions by unrelated objects or a limited field of view) [68]. Consider a simple example sequence shown in Fig. 1 from the EPIC-Kitchens dataset [11]. Although this is a first-person video with a limited point of view, we as human beings can easily infer the action after *take bottle* and *pour water* to be *drink water*, even though the drinking action is not directly observed. This is because our brains can reason about the relation of actions: a drink water action should have happened since we first see the camera wearer takes the bottle, fills the glass with water, and then observed he/she puts down the empty glass. Because of the limited observation, it is difficult for existing methods based on convolutional neural networks to perform well [68].

In this work, we use Graph Convolutional Networks (GCNs) [12, 30] as a key tool and propose a novel model called Graph-based Temporal Reasoning Module (GTRM) that can be built on top of existing action segmentation models (*backbone models*) to predict better action segmentation by learning the temporal relations among actions. Given an initial action segmentation result of the backbone model, we map each segment to a graph node then construct two graphs for refining the classification and the temporal boundary of each node. By jointly optimizing the backbone model and the proposed model, we can explicitly model the relation of neighboring actions and thus refine the segmentation result. Furthermore, since a node represents an action segment of arbitrary length, the GCNs operate on a flexible temporal receptive field, which makes it easier to capture both short and long-range temporal relations.

The effectiveness of our model is evaluated on two datasets: the EGTEA dataset [40] and the EPIC-Kitchens dataset [11]. We choose these datasets for two reasons. Firstly, action segmentation in egocentric videos of the two datasets is more challenging than in videos captured from a fixed, third-person point of view. This is because many actions may not be directly observable due to the limited field of view and severe occlusions caused by the camera wearer's hand or other objects. Secondly, the datasets contain long videos (*e.g.* $>$ 10 min) with many action instances (*e.g.* $>$ 100), making it difficult for existing action segmentation models to work properly. Experiments on the two datasets demonstrate that our GTRM can largely improve the performance of the backbone models. We additionally show by experiments that our model works better with backbone models using recurrent networks. Moreover, we demonstrate that our proposed model can also improve the backbone performance on general third-person datasets for action segmentation, *i.e.* 50Salads [54] and Breakfast [32]. To summarize, the main contributions of this work are:

- To the best of our knowledge, this work takes the first step towards explicitly leveraging the relations among more than two actions for action segmentation.
- We construct graphs using initial action segments and establish edges to model the relation of the segments. By applying GCNs on the graph, the node representation can be updated based on the relations with its neighbors to predict a better action segmentation.
- Experiments on multiple datasets show the effectiveness of our GTRM for improving action segmentation of multiple state-of-the-art backbone models.

2. Related work

Action segmentation Unlike action detection methods which output a sparse set of action segments, action segmentation methods predict what action is occurring at every frame in a video [35]. Because of the wide range of potential applications, action segmentation has long been studied by many researchers [4, 13, 21, 35]. For example, the works by Fathi *et al*. modeled actions by the state change of objects being manipulated and used a segmental model to learn a set of temporally consistent actions [17–19]. Cheng *et al*. [10] used bag of visual words as representations of videos and use a hierarchical Bayesian non-parametric model for segmenting events in videos. However, the optimization of these works is typically slow, especially for long videos. A line of work [14, 34, 50, 52] focuses on the task of weakly supervised action segmentation, while the assumption that there exists a strict ordering of actions which is not applicable in general cases.

To ensure temporal smoothness of action segments, many approaches apply temporal classifiers over frame-wise features. Several works [33, 55, 57] used probabilistic models to predict the most probable sequence of action. Lea *et al*. [35] first proposed to use temporal convolution networks (TCN) for action segmentation. They have proved that TCNs can outperform traditional sliding window based methods [29, 51]. Lei *et al*. [37] further equipped TCN with deformable convolution and residual stream. However, these two models [35, 37] only works on a low temporal resolution. Recently, Farha *et al*. [16] proposed to use dilated TCN with multi-stage refinement to capture information from a large temporal receptive field. The dilated convolution avoids using temporal pooling to capture long-range dependencies and thus could operate on full temporal resolution and achieve the state-of-the-art performance. However, none of the existing methods explicitly leverage the relations among more than two actions to enhance action segmentation. Due to the lack of relational reasoning ability, it is still hard for existing methods to capture the actions that are not directly observed [68].

To address this problem, our model learns the relation of actions by constructing the segments as nodes and applying graph convolution networks. The representation of each node is aggregated from other nodes through the graph edges and thus the relation among actions is leveraged to achieve a better action segmentation result.

Graph convolution networks After proposed in [30], graph convolution networks (GCNs) have been proved to be effective in modeling the relation of data with non-grid structures [39, 41]. Since then, GCNs have shown convincing successes in modeling relations [2, 15, 22] and thus are widely applied in multiple research tasks like semi-supervised learning [38], image captioning [63], skeleton-based action recognition [62], and video action recognition [58, 60, 65, 66]. For instance, Pan *et al*. [44] applied GCN to model the relation of human joints for the task of

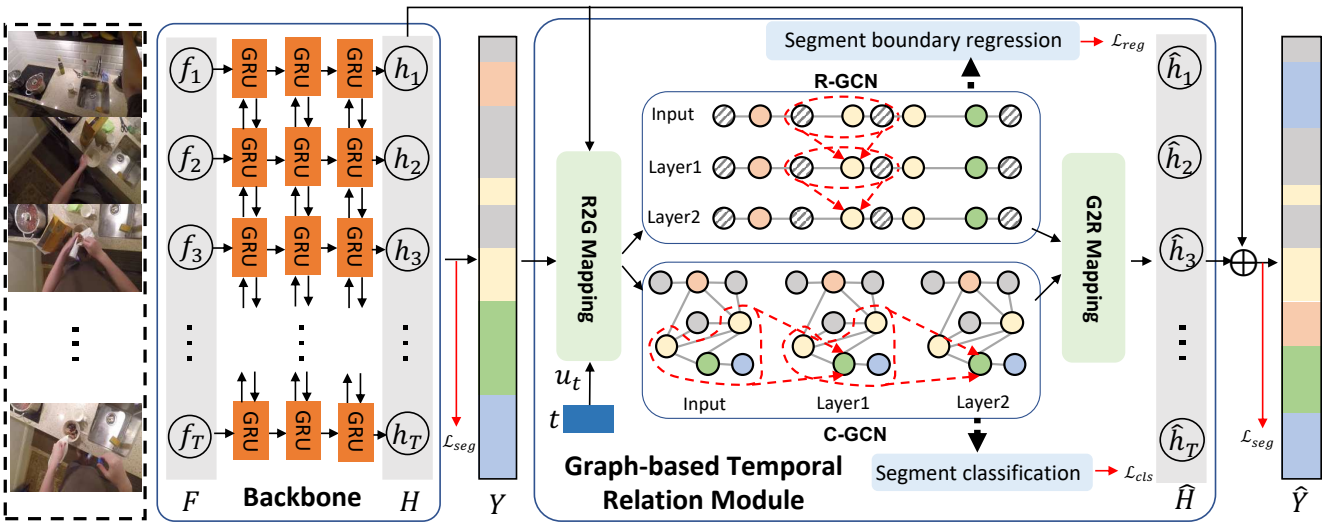

Figure 2. Illustration of our proposed Graph-based Temporal Relation Module (GTRM) built on top of a 3-layer GRU backbone model. Our GTRM maps the encoded representation of each segment in the initial segmentation to a node in the graph. The two graphs have different edges and correspond to two target tasks of segment boundary regression and segment classification. The node representations updated by GCNs are mapped back to frame-wise representations for a finer action segmentation.

action assessment. Zeng *et al*. [64] proposed a model to consider relations of multiple action proposals for more accurate action localization. Our GTRM is inspired by these works, and we exploit the ability of GCNs to explicitly model the temporal relations of actions for video action segmentation.

3. Graph-based Temporal Reasoning Module

Given a video of a total T frames, our goal is to infer the action class label of each frame, whose ground-truth is given by $Y^{gt} = \{\boldsymbol{y}_1^{gt}, \cdots, \boldsymbol{y}_T^{gt}\}$, where $\boldsymbol{y}_t^{gt} \in \{0,1\}^C$ is a one-hot vector where the true class is 1 and others are all 0. C is the number of classes including the background class meaning no action. Our GTRM is built on top of a backbone model for action segmentation and refines the original estimation result through graph-based reasoning.

In the following, we explain the details of our GTRM and its training process. We denote a graph by $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ are a set of N nodes and $e(i,j) \in \mathcal{E}$ represents the weight of the edge connecting the nodes i and j. The implementation details are given at the end of this section.

3.1. Overview

The architecture of our GTRM is illustrated in Fig. 2. We show a 3-layer GRU as an example for the backbone model, but it can be generalized as a model that takes input frame-wise features $F = \{\boldsymbol{f}_1, \cdots, \boldsymbol{f}_T\}$ extracted using some feature extractors, *e.g*. I3D [6] and outputs the initial action class likelihoods $Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_T\}$ where $\boldsymbol{y}_t \in [0,1]^C$. Our GTRM takes Y as input, together with the frame-wise d-dimensional hidden representations $H = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_T\}$ encoded by the backbone model.

Inspired by the recent success on relational reasoning [9, 25, 27, 64], we build our model using GCNs for learning temporal relations of actions. We first construct two graphs, called R-GCN and C-GCN, by mapping the hidden representations H from the backbone model to the graph nodes. Each node of the graph represents each action segment (*i.e*. consecutive predictions in Y with the highest likelihood on the same action category), and graph edges represent the relation between the two corresponding action segments.

Each graph is associated with a different loss function during the training process, *i.e*. a segment boundary regression loss for R-GCN and a segment classification loss for C-GCN, and different sets of edges to account for these tasks. Graph convolutions are performed separately on R-GCN and C-GCN to update node representations by aggregating information from neighboring nodes.

We map the updated node representations back to form an updated frame-wise representation $\hat{H}$, and combine with the backbone representation H to predict a better frame-wise segmentation. The loss function over the segmentation outputs and the loss functions for each of the GCN are used to jointly train the backbone model and the GTRM. The details of the proposed GTRM will be given in the following sections.

3.2. Representation-to-Graph (R2G) Mapping

The key step in our proposed model is to construct the graphs based on the action class likelihood Y and the hidden

representation H of the backbone model. We call this step Representation-to-Graph (R2G) mapping, since the graph node representations are mapped from the output representation H of the backbone model. Suppose we have a total of N temporally ordered segments in Y. The i-th action segment can be represented as $(t_{i,s}, t_{i,e})$, in which $t_{i,s}$ and $t_{i,e}$ are the starting and ending frames of the action segment, respectively. Each node of R-GCN and C-GCN corresponds to such initial action segment, and the hidden representation $\boldsymbol{a}_i$ of each node is obtained by applying max pooling over the set of hidden representations corresponding to the action segment $\{\boldsymbol{h}_{t_{i,s}}, \cdots, \boldsymbol{h}_{t_{i,e}}\}$. In addition, since the temporal location of each segment contains useful information such as ordering, we also encode the time information to a d_t-dimensional vector $\boldsymbol{u}_i$ by feeding the time vector $(t_{i,s}, t_{i,e})$ to a multi-layer perceptron. The representation $\boldsymbol{x}_i$ for the i-th node is obtained by concatenating $\boldsymbol{a}_i$ and $\boldsymbol{u}_i$ in a channel-wise manner.

Defining fully connected graph edges to model the temporal relations of all action segments [60] can potentially result in noisy message passing between unrelated actions that are temporally far apart. To better address the action segmentation task, which essentially can be viewed as finding the class label and temporal boundary of all action instances including the background (no action), we construct different types of edges for the two graphs where the edges of R-GCN correspond to the boundary regression task and the edges of C-GCN to the classification task.

R-GCN The target task of the R-GCN is segment boundary regression, and its edges are defined to model the relation between neighboring segments which directly determine the temporal boundary (*i.e.* the start and the end frames) of the corresponding action segment. To this end, we only connect each segment with the segments right next to it by computing the temporal proximity between two segments. Defining $p(i,j)$ as the temporal proximity (inverse of the distance) between the middle frames of the i-th and j-th segment normalized by the length of the video, the edges $e_r(i,j)$ between the i-th and j-th nodes in R-GCN are defined as

$$e_r(i,j) = \begin{cases} p(i,j) & |i-j| \leq 1 \\ 0 & otherwise. \end{cases} \tag{1}$$

C-GCN In contrast, the target task of the C-GCN is segment classification, and the edges have to take into account the relations among multiple actions as they influence or condition on each other. For example, if we see a *take knife* action and then a *take potato* action, it is highly likely that a *cut potato* action will happen in the next few segments. We can infer the *cut potato* action even when the potato is occluded by leveraging such temporal relations. However, if two actions have a long temporal gap, they are unlikely to influence each other. Thus, we define edges $e_c(i,j)$ in C-GCN based on temporal proximity between the two nodes as

$$e_c(i,j) = \begin{cases} p(i,j) & |j-i| \leq 1, c_i \vee c_j = bg \\ p(i,j) & |j-i| \leq k, c_i \neq bg, c_j \neq bg \\ 0 & otherwise, \end{cases} \tag{2}$$

where bg represents the background class where no action happens. In other words, each background node is linked only to its nearest neighbors, while each of other nodes is also linked to k neighboring nodes.

3.2.1 Reasoning on Graphs

In both GCNs, all of the edge weights form the adjacency matrix $\boldsymbol{A}_c$ or $\boldsymbol{A}_r$ with $N \times N$ dimensions. Following [60], we normalize the adjacency matrix by using the softmax function as

$$\boldsymbol{A}(i,j) = \frac{\exp g(i,j)}{\Sigma_{j=1}^{N} \exp g(i,j)}. \tag{3}$$

For reasoning on the graphs, we perform M-layer graph convolution for refining the node representation. Graph convolution enables message passing based on the graph structure, and multiple GCN layers further enable message passing between non-connected nodes [30]. In an M-layer GCN, the graph convolution operation of the m-th layer $(1 \leq m \leq M)$ could be represented as

$$\boldsymbol{X}^{(m)} = \sigma(\boldsymbol{A}\boldsymbol{X}^{(m-1)}\boldsymbol{W}^{(m)}), \tag{4}$$

where $\boldsymbol{X}^{(m)}$ are the hidden representation of all the nodes with $N \times d_m$ dimensions at the m-th layer. $\boldsymbol{W}^{(m)}$ is the weight matrix of the m-th layer, and σ denotes the activation function. Following prior work [60], we apply two activation functions namely Layer Normalization [1] and ReLU after each GCN layer. After the graph convolution operations, we obtain updated node representations $\hat{\boldsymbol{x}}_i^c$ and $\hat{\boldsymbol{x}}_i^r$ for nodes in the C-GCN and R-GCN, respectively.

We apply an FC layer on each node after the final GCN layer to perform segment classification on the C-GCN and segment boundary regression on the R-GCN. This operation is also known as *readout* operation [48, 59] as it maps the refined node representation to the desired output. The output of each C-GCN node is the class likelihood $\hat{\boldsymbol{c}}_i$ for the corresponding segment. Following previous works on boundary regression [20, 49], the output of each node in R-GCN is an offset vector $\hat{\boldsymbol{o}} = (\hat{o}_{i,c}, \hat{o}_{i,l})$ relative to the input segment. $\hat{o}_{i,c}$ is the offset of the segment center (normalized by the length of the segment), and $\hat{o}_{i,l}$ is the offset of the length of a segment in log scale. Given these offsets, it is trivial to compute the predicted boundary $\hat{t}_{i,s}, \hat{t}_{i,e}$.

3.3. Graph-to-Representation (G2R) Mapping

After the graph convolution operations, the representation of each node is updated by information propagation from its neighboring nodes. To perform action segmentation based on the updated representations, we inversely map the updated graph node representations to frame-wise representations $\hat{H} = \{\hat{\boldsymbol{h}}_1, \cdots, \hat{\boldsymbol{h}}_T\}$. We fuse the representations from two GCNs via node-wise summation, and then reconstruct $\hat{\boldsymbol{h}}$ by mapping the node representation to all of the corresponding frames:

$$\hat{\boldsymbol{h}}_t = \hat{\boldsymbol{x}}_i^c + \hat{\boldsymbol{x}}_i^r, \ \forall t \in \{\hat{t}_{i,s}, \cdots, \hat{t}_{i,e}\}, \tag{5}$$

where $\hat{t}_{i,s}, \hat{t}_{i,e}$ are the temporal starting and ending frames of the i-th segment predicted by the R-GCN. Similarly to previous work [64, 67], we concatenate $\hat{\boldsymbol{h}}$ with the original latent representation $\boldsymbol{h}$ from the backbone model for obtaining the final action segmentation results. We apply a 1×1 convolution layer on the concatenated representation followed by softmax as activation function to obtain the final frame-wise action likelihood $\hat{\boldsymbol{y}}$.

3.4. Training and Loss Function

We train the whole network including both the backbone model and our GTRM using a combination of multiple loss functions. As for the action segmentation outputs $\boldsymbol{y_t}, \hat{\boldsymbol{y}}_t$, we apply the same loss function as [16] which is a combination of cross entropy loss $\mathcal{L}_{\text{cls}}$ and a truncated mean squared error $\mathcal{L}_{\text{t-mse}}$ designed to punish local inconsistency by encouraging adjacent predictions to be similar:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{cls}} + \lambda_t \mathcal{L}_{\text{t-mse}}. \tag{6}$$

We use the same cross entropy loss $\mathcal{L}_{\text{cls}}$ for C-GCN. The ground truth action category of a segment is defined by the category of the closest ground truth segment measured by temporal intersection over union (tIoU).

For R-GCN, we use smooth L1 loss as the regression loss $\mathcal{L}_{\text{reg}}$. Similarly with the C-GCN, the ground truth time information of a node is defined by the temporally closest segment to this node. Denote $t_{i,c} = (t_{i,s} + t_{i,e})/2$ and $t_{i,l} = t_{i,e} - t_{i,s}$ as the center and length of a segment, respectively, the ground truth offset $\boldsymbol{o}_i^{gt} = (o_{i,c}^{gt}, o_{i,l}^{gt})$ could be represented as:

$$o_{i,c}^{gt} = (t_{i,c} - t_{i,c}^{gt})/t_{i,l}, \quad o_{i,l}^{gt} = \log(t_{i,l}/t_{i,l}^{gt}), \tag{7}$$

The combined loss function thus can be defined as

$$\begin{aligned} \mathcal{L} = &\sum_{i=1}^{T} \mathcal{L}_{\text{seg}}(\boldsymbol{y}_i^{gt}, \boldsymbol{y}_i) + \sum_{i=1}^{T} \mathcal{L}_{\text{seg}}(\boldsymbol{y}_i^{gt}, \hat{\boldsymbol{y}}_i) \\ &+ \lambda_1 \sum_{i=1}^{N} \mathcal{L}_{\text{cls}}(\boldsymbol{c}_i^{gt}, \hat{\boldsymbol{c}}_i) + \lambda_2 \sum_{i=1}^{N} \mathcal{L}_{\text{reg}}(\boldsymbol{o}_i^{gt}, \hat{\boldsymbol{o}}_i). \end{aligned} \tag{8}$$

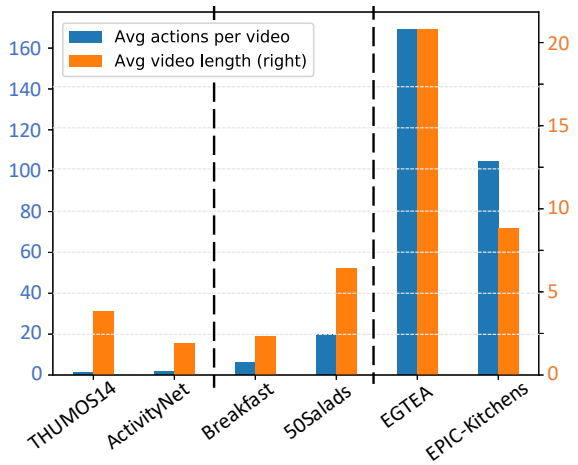

Figure 3. Dataset comparison by average action instances per video (blue) and average video length (orange, right axis).

3.5. Implementation Details

We implement our model using the Pytorch [45] library. We choose to use $d = 64$ as the dimension of hidden representations. The multi-layer perceptron for encoding the time representation $\boldsymbol{u}_t$ is a fully connected layer with sigmoid activation and 16 output channels. We use 2 layer GCNs in all of our experiments, since we do not observe obvious performance increase when adding more layers. The details about training can be found in the supplementary material.

4. Experiments

We compare the performance of our action segmentation model with state-of-the-art models on challenging large-scale datasets. We also conduct ablation studies to examine the impact of each part of our model, and we examine the performance of our GTRM when built on top of existing backbone models on more general third-person datasets.

Datasets Figure 3 compares different commonly-used video datasets based on average action instances per video and average video length (in minutes), in which we divide them into three groups. The leftmost group are the THUMOS14 [28] and ActivityNet [5] dataset. These datasets contain one or two action instances per video and are usually used for the task of action proposal [42], localization [47] or detection [43, 61]. The Breakfast [32] and 50Salads [54] dataset contain less than 20 actions per video, and are the standard datasets for evaluation of action segmentation methods [37]. The rightmost group contains two recent large-scale datasets containing natural daily living activities from an egocentric perspective, EGTEA [40] and EPIC-Kitchens [11]. Due to the unique perspective of egocentric recording, the actions sometimes happen out of the

camera's field of view (*e.g.* in Fig. 1), or critical informative region is occluded by the hand. These characteristics make many actions in EGTEA and EPIC-Kitchens not directly observable and they have to be inferred from temporal relations. In the following sections, we mainly conduct the experiments on these two datasets, while later we also show experimental results on the Breakfast and 50Salads datasets.

Evaluation Metrics For evaluating our model, we adopt several evaluation metrics commonly used in action segmentation [16, 35, 37]: frame-wise accuracy, segmental edit score, and the segmental F1 score at overlapping thresholds $\tau/100$ denoted by F1@τ. Frame-wise accuracy is one of the most widely used metrics for evaluation of action segmentation. However, long actions tend to have a higher impact on this metric, while there is no strong penalty on over-segmentation. In contrast, segmental edit score and F1 score are evaluation metrics presented in [35, 36] and penalize over-segmentation errors. The segmental edit score penalizes the case of over-segmentation, and the segmental F1 scores measure the quality of the prediction.

4.1. Comparison with the State of the Art

In this section, we compare our model with several state-of-the-art models on EGTEA and EPIC-Kitchens datasets (Table 1). The EGTEA dataset contains 86 videos and has a total length of 29 hours. We focus on the segmentation of the 19 action classes (*i.e.* verbs). For the EGTEA dataset we perform a four-fold cross validation by randomly splitting the videos into four partitions. The EPIC-Kitchens dataset contains 55 hours of daily living non-scripted activities with 125 classes of actions. Since the ground-truth labels of the test set are not publicly available, we follow [3] to split part of the training set as train-test set. The video features for EGTEA and EPIC-Kitchens are extracted by using I3D pretrained on Kinetics dataset [6]. We down-sample the videos to 15 fps.

We use four closely related methods as baseline models. **FC** is a simple baseline that directly add a frame-wise classifier on the I3D-extracted features. **Bi-LSTM** [53] is a bi-directional temporal LSTM for action segmentation. **EDTCN** [35] and **MSTCN** [16] are two of the recent competitive models using temporal convolution networks to capture long term frame dependencies. We also include our own backbone using multi-layer GRU (**m-GRU**) in the comparison. We report the performances of our GTRM built on top of different backbone networks, by adding "+GTRM " as the notation. Since no previous results on EGTEA and EPIC-Kitchens datasets are available for baseline models, all the reported results are based on our implementation.

As can be seen from and Table 1, comparing our model with the backbone models (without adding our GTRM), our model outperforms backbone models by a large margin on F1 score and edit score, while performing comparably well with respect to the frame-wise accuracy metric. The lower parts of Table 1 summarize the performance of our proposed GTRM when built on top of different backbones. As can be seen, the performance of all backbone models mostly increases by adding GTRM, except the F1@50 and accuracy of MSTCN in the EGTEA dataset. This shows that our GTRM is capable of refining the backbone results in most cases. Interestingly, we find that the gain of adding our GTRM is the largest with recurrent backbone models (Bi-LSTM and m-GRU). This is possibly because the recurrent backbones have a smaller span of attention, while our GTRM can work complementary since the reasoning is performed with a larger temporal receptive field.

From the qualitative comparison in Fig. 4 (a), we can see that the "take", "put" and "close" actions are correctly detected by adding our GTRM. Especially, due to the viewpoint limitation, the "close (fridge)" action is almost not observable in the video (since the camera wearer quickly turns his attention to the location of the next step). The fact that this action is being correctly detected by our model strongly supports our claim that our GTRM can capture the relation of actions (as there is an "open (fridge)" action happened before) for better action segmentation. On the other hand, we can also see weakness of our model in Fig. 4 (b) is that our GTRM depends on the initial backbone output. The backbone model could not detect the "read" action, and the "take", "put" actions are predicted as a single "cut" action. Conditioning on this output, it is still difficult for our

EGTEA	**F1@{10,25,50}**			**Edit**	**Acc**
FC [6]	8.7	6.7	3.1	9.4	65.4
Bi-LSTM [53]	27.0	23.1	15.1	28.5	70.0
EDTCN [35]	31.1	27.7	19.6	28.6	70.1
MSTCN [16]	32.1	28.3	18.9	32.2	69.2
m-GRU	32.6	27.7	17.6	36.0	67.1
Bi-LSTM+GTRM	33.3	29.2	19.9	32.1	**70.7**
EDTCN+GTRM	34.6	31.2	20.7	34.8	70.1
MSTCN+GTRM	36.6	29.7	18.6	32.2	68.4
m-GRU+GTRM	**41.6**	**37.5**	**25.9**	**41.8**	69.5

EPIC	**F1@{10,25,50}**			**Edit**	**Acc**
FC	9.3	5.6	2.2	20.0	42.2
Bi-LSTM [53]	19.0	11.7	5.0	29.1	43.3
EDTCN [35]	21.8	13.8	6.5	27.3	42.9
MSTCN [16]	19.4	12.3	5.7	25.3	43.6
m-GRU	20.2	15.2	7.7	30.5	40.3
Bi-LSTM+GTRM	25.1	17.3	8.8	35.9	43.5
EDTCN+GTRM	24.2	15.9	7.2	33.1	42.8
MSTCN+GTRM	24.4	15.4	7.2	32.5	**43.7**
m-GRU+GTRM	**31.9**	**22.8**	**10.7**	**42.1**	43.4

Table 1. Quantitative comparison with state-of-the-art models on the EGTEA dataset (top) and EPIC-Kitchens dataset (bottom).

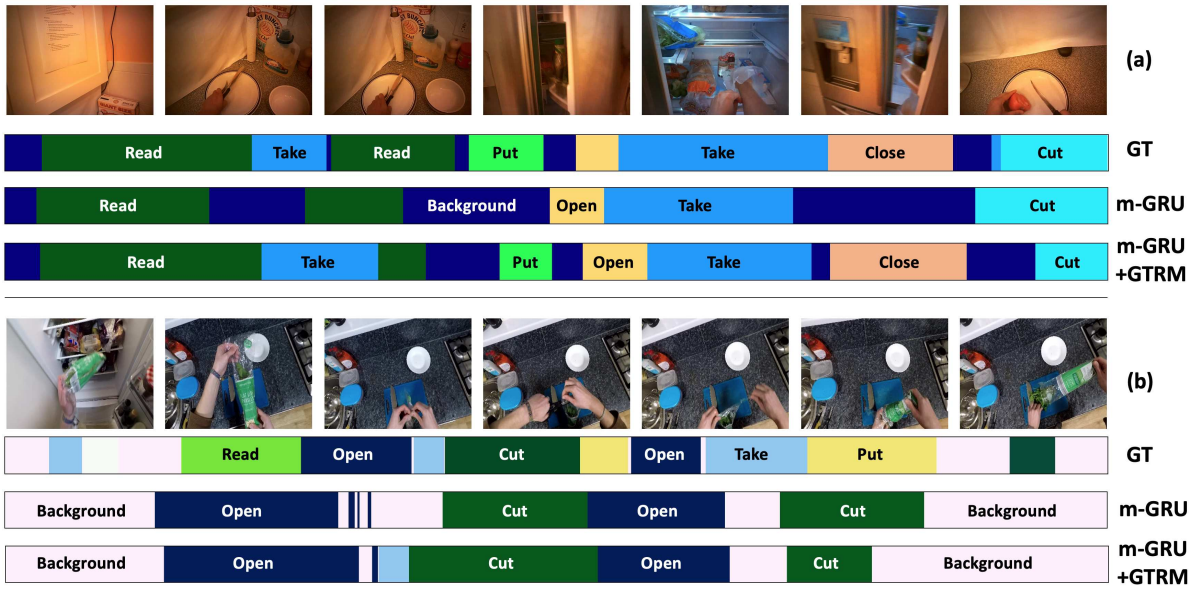

Figure 4. Qualitative comparison of results for action segmentation task on (a) EGTEA, and (b) EPIC dataset. Only part of the whole video is shown for clarity. We can see in (a) that the *take*, *put* and *close* actions are correctly detected by adding GTRM.

GTRM to correctly identify those actions. More qualitative results of different backbone models with and without our proposed GTRM are in the supplementary material.

4.2. Ablation Studies

To fully understand the effect of each component of our model, we conduct ablation studies on the EGTEA dataset by changing or deleting part of our model and compare their performances. We first examine the impact of each of the graphs in our model. For fair comparison, we replace each of the C-GCN and R-GCN with a small 2 layer fully connected network (denoted by FCN). In this case, each graph node is processed individually by the FCN without considering the relations brought by the graph edges. We also examine the usefulness of time vector $\boldsymbol{u}_t$. Table 2 shows the *relative performance gain* compared with using the m-GRU backbone alone. In the table, C-GCN + FCN is the case where R-GCN is replaced by the fully connected network and others follow the same rule. We can see that the performance using GCN in general favors than that without GCN, which validates the usefulness of using relations between actions for action segmentation. Additionally, we find that the time vector $\boldsymbol{u}_t$ provides necessary information to the network as adding $\boldsymbol{u}_t$ improves the performance while without $\boldsymbol{u}_t$ the task for boundary regression cannot converge.

We also investigate the selection of parameter k, which is related to the number of neighbors for each segment to aggregate information from. We variant the value of k and show the experiment result on EGTEA dataset in Fig. 5. Overall, the best performance is achieved with $k = 8$, while the performance gain decreases starting from $k = 16$. We suspect this is because of irrelevant information propagation through the edges by connecting the action segments that are too temporally distinct. Further ablation studies on the influence of edge weight and tools for modeling relation (*e.g.* 1D convolution on nodes) can be found in the supplementary material.

Gain	F1@{10,25,50}			Edit	Acc
C-GCN + FCN (w/o $\boldsymbol{u}_t$)	4.6	4.4	1.8	4.4	2.5
FCN only	6.2	6.1	4.7	4.5	3.3
R-GCN + FCN	6.8	6.8	4.8	6.0	2.1
C-GCN + FCN	6.4	6.0	4.7	3.5	2.7
C-GCN + R-GCN	10.0	9.8	6.8	7.5	2.8

Table 2. Ablation study of our model. We replace GCN with fully connected network (FCN) and report the performance gain in absolute values relative to the m-GRU backbone model.

4.3. Results on Other Datasets

To test the effectiveness of our proposed model on other general cases, we also test our model performance on the 50Salads [54] and Breakfast [32] datasets. The 50Salads dataset contains 50 videos of salad making activities with 17 action classes. We follow [54] to use a 5-fold cross validation and report the average performance. The Breakfast dataset contains 1712 videos with a total length of 65 hours. There are 48 different actions while on average 6 actions per video. We use the standard 4 splits [32] and report the average. For a fair comparison, we adopt the features from [16] in the following experiments.

We build our GTRM on top of the current state-of-the-art

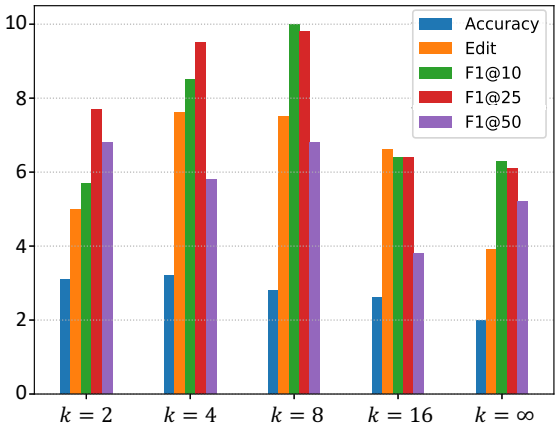

Figure 5. Performance gain compared with the m-GRU backbone model with different values of k. $k = \infty$ denotes the case that all nodes are connected.

50Salads	**F1@{10,25,50}**			**Edit**	**Acc**
MSTCN [16]	76.3	74.0	64.5	67.9	80.7
MSTCN(our impl.)	73.4	71.0	61.5	67.2	80.2
MSTCN+GTRM	75.4	72.8	63.9	67.5	82.6
Gain	2.0	1.8	1.4	0.3	2.4
Bi-LSTM [53]	62.6	58.3	47.0	55.6	55.7
Bi-LSTM (out impl.)	62.2	61.3	53.7	53.5	70.1
Bi-LSTM+GTRM	70.4	68.9	62.7	59.4	81.6
Gain	8.2	7.6	9.0	5.9	11.5

Table 3. Results on the 50 Salads dataset. Performance gain in absolute values by adding our GTRM on top is shown in dark rows.

approach MSTCN [16]. Since MSTCN is based on temporal convolution networks, we further test the model performance combined with a recurrent backbone Bi-LSTM [53]. The performance comparison on 50Salads dataset is shown in Table 3, including both the result reported in [16] and result with our implementation. Since there are on average 20 actions per video, we adjust the parameter k to be 4. As can be seen, the performance of both backbone models got improved by adding our GTRM. While the performance gain of the MSTCN backbone is relatively marginal, the gain of Bi-LSTM backbone is still significant. This phenomenon is the same as observed in the EGTEA dataset, which shows that our GTRM works better with recurrent backbones.

Since there was no previously reported results from Bi-LSTM, we only use MSTCN as the backbone model for the Breakfast dataset. The performance is summarized in Table 4. The breakfast dataset only contains 6 action instances per video, far less than the 50Salads dataset. Similarly with the 50Salads dataset, the performance gain is relatively marginal. Also, modeling relations among more neighbors by increasing k does not improve the segmentation performance.

Breakfast	**F1@{10,25,50}**			**Edit**	**Acc**
MSTCN [16]	52.6	48.1	37.9	61.7	66.3
MSTCN (our impl.)	57.3	53.4	41.4	58.8	60.0
MSTCN+GTRM ($k = 2$)	57.5	54.0	43.3	58.7	65.0
Gain	0.2	0.6	1.9	-0.1	5.0
MSTCN+GTRM ($k = 4$)	57.3	53.6	42.9	58.5	63.8
Gain	0.0	0.2	1.5	-0.3	3.8

Table 4. Result on the Breakfast dataset. Performance gain in absolute values by adding our GTRM on top is shown in dark rows.

There could be mainly two reasons why the benefit of our GTRM is limited on these two datasets. Firstly, as the 50Salads and Breakfast dataset are taken from a fixed view camera capturing most of the human activities, there are less cases of unobservable actions due to, *e.g.*, occlusions. Secondly, the number of action instances is relatively small so that temporal patterns can be captured to some extent by only using the backbone model.

4.4. Limitations and Future Work

As discussed in Section 4.1, one of the limitations of our model is that it relies on the backbone model. If the backbone model output a poor result, our model can only slightly improve the segmentation performance.

Another limitation is that, if the backbone outputs are heavily fragmented, the constructed graph would be large and the optimization becomes very inefficient. This also prevents us from building our model on top of the FC baseline. While it is possible to filter the action segments and ignore the small segments in the graph construction step, it is still an important future work to examine approaches to process the graph convolution in a more efficient way. Using additional information like gaze [26] or techniques such as adaptive sampling [24] or stochastic training [8] will be promising candidates for future investigation.

5. Conclusion

In this paper, we presented a novel approach for modeling action relations aiming at the task of action segmentation which can be built on top of most existing neural networks for action segmentation. To model the temporal relations, we construct two graphs and use GCNs to perform reasoning on the graphs based on two different criteria. After updating the node representations, they are mapped back to individual frames as an updated representation for final action segmentation. Extensive experiments showed that our model can effectively learn to use relations for better action segmentation, and demonstrated performance improvements brought by our model.

Acknowledgement This work is supported by JST CREST Project and the GCL program of the University of Tokyo.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] Mohit Bajaj, Lanjun Wang, and Leonid Sigal. G3raphground: Graph-based language grounding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[3] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 6

[4] Subhabrata Bhattacharya, Mahdi M Kalayeh, Rahul Sukthankar, and Mubarak Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *roceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[6] 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 (CVPR)*, 2017. 1, 3, 6

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 2009. 1

[8] Jianfei Chen, Jun Zhu, and Le Song. Stochastic training of graph convolutional networks with variance reduction. *arXiv preprint arXiv:1710.10568*, 2017. 8

[9] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[10] Yu Cheng, Quanfu Fan, Sharath Pankanti, and Alok Choudhary. Temporal sequence modeling for video event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 5

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems (NeurIPS)*, 2016. 2

[13] Li Ding and Chenliang Xu. Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation. *arXiv preprint arXiv:1705.07818*, 2017. 2

[14] 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 (CVPR)*, 2018. 1, 2

[15] Lifeng Fan, Wenguan Wang, Siyuan Huang, Xinyu Tang, and Song-Chun Zhu. Understanding human gaze communication by spatio-temporal graph reasoning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[16] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5, 6, 7, 8

[17] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 2

[18] Alireza Fathi and James M Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[19] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[20] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 4

[21] Pallabi Ghosh, Yi Yao, Larry S Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. *arXiv preprint arXiv:1811.10575*, 2018. 2

[22] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. *arXiv preprint arXiv:1905.04405*, 2019. 2

[23] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1

[24] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 8

[25] Yifei Huang, Minjie Cai, Hiroshi Kera, Ryo Yonetani, Keita Higuchi, and Yoichi Sato. Temporal localization and spatial segmentation of joint attention in multiple first-person videos. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (CVPRW)*, 2017. 3

[26] Yifei Huang, Minjie Cai, Zhenqiang Li, and Yoichi Sato. Predicting gaze in egocentric video by learning task-dependent attention transition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 8

[27] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. Videograph: Recognizing minutes-long human activities in videos. *arXiv preprint arXiv:1905.05143*, 2019. 3

[28] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 2017. 5

[29] Svebor Karaman, Lorenzo Seidenari, and Alberto Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCV THUMOS Workshop*, 2014. 1, 2

[30] Thomas N Kipf and Max Welling. Semi-supervised classi-

fication with graph convolutional networks. *ICLR*, 2017. 2, 4

[31] Volker Krüger, Danica Kragic, Aleš Ude, and Christopher Geib. The meaning of action: A review on action recognition and mapping. *Advanced robotics*, 2007. 1

[32] 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 (CVPR)*, 2014. 2, 5, 7

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

[34] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 2017. 2

[35] 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 (CVPR)*, 2017. 1, 2, 6

[36] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1, 6

[37] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6742–6751, 2018. 1, 2, 5, 6

[38] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[39] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[40] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 5

[41] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[42] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[43] Khoi-Nguyen C. Mac, Dhiraj Joshi, Raymond A. Yeh, Jinjun Xiong, Rogerio S. Feris, and Minh N. Do. Learning motion in feature space: Locally-consistent deformable convolution networks for fine-grained action detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[44] Jia-Hui Pan, Jibin Gao, and Wei-Shi Zheng. Action assessment by joint relation graphs. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[45] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[46] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[47] Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Hierarchical self-attention network for action localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[48] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 4

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 4

[50] 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 (CVPR)*, 2017. 2

[51] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2

[52] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[53] 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 (CVPR)*, 2016. 1, 6, 8

[54] 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*, 2013. 2, 5, 7

[55] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[56] Robin R Vallacher and Daniel M Wegner. What do people think they're doing? action identification and human behavior. *Psychological review*, 1987. 1

[57] Nam N Vo and Aaron F Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2014. 2

[58] Shaojie Wang, Wentian Zhao, Ziyi Kou, and Chenliang Xu. How to make a blt sandwich? learning to reason towards understanding web instructional videos. *arXiv preprint arXiv:1812.00344*, 2018. 2

[59] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J. Cran-

dall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 4

[60] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 4

[61] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S. Davis, and David J. Crandall. Temporal recurrent networks for online action detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[62] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[63] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[64] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 3, 5

[65] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *arXiv preprint arXiv:1908.09995*, 2019. 2

[66] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[67] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2017. 5

[68] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 2