

# OTAS: Unsupervised Boundary Detection for Object-Centric Temporal Action Segmentation

Yuerong Li <sup>1</sup>, Zhengrong Xue <sup>2,3,4</sup>, Huazhe Xu <sup>2,3,4</sup>

<sup>1</sup> Zhejiang University <sup>2</sup> Tsinghua University <sup>3</sup> Shanghai Qi Zhi Institute <sup>4</sup> Shanghai AI Lab

<https://github.com/y1596/OTAS>

## Abstract

Temporal action segmentation is typically achieved by discovering the dramatic variances in global visual descriptors. In this paper, we explore the merits of local features by proposing the unsupervised framework of Object-centric Temporal Action Segmentation (OTAS). Broadly speaking, OTAS consists of self-supervised global and local feature extraction modules as well as a boundary selection module that fuses the features and detects salient boundaries for action segmentation. As a second contribution, we discuss the pros and cons of existing frame-level and boundary-level evaluation metrics. Through extensive experiments, we find OTAS is superior to the previous state-of-the-art method by 41% on average in terms of our recommended F1 score. Surprisingly, OTAS even outperforms the ground-truth human annotations in the user study. Moreover, OTAS is efficient enough to allow real-time inference.

## 1. Introduction

Temporal action segmentation [7] aims to label every frame in an untrimmed video with action tags. With the emergence of large-scale instructional video datasets [2, 17, 39], various learning-based frameworks [6, 17, 31] tackle action segmentation under supervision. However, densely-annotated video datasets are often criticized to be excessively expensive; even worse, human-decided action boundaries are subjective, leading to non-negligible biases.

In the more tempting fully unsupervised scenario where the action types are unavailable, a set of predicted action boundaries could sufficiently infer the untagged temporal segmentation. Thus, there appear two branches of works: one branch [21, 24, 34] learns to group similar frames and transforms the segmentation problem into a clustering process; the other branch [1, 10, 48] learns to detect the boundaries that indicate the most salient variations among the frames and generates segmentation afterwards.

While the current research hotspots for unsupervised ac-

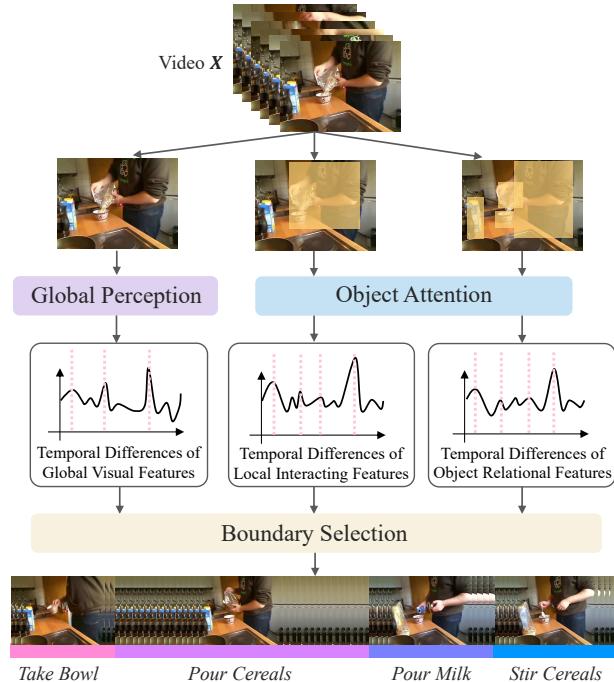


Figure 1. **The framework of Object-centric Temporal Action Segmentation (OTAS).** Given some untrimmed sequences of frames, OTAS learns global visual features, local interacting features, and object relational features via the self-supervised global perception and object attention modules. Afterwards, the boundary selection module fuses the generated global and local features to detect the salient boundaries for action segmentation.

tion segmentation are mainly the elaborately designed clustering or boundary selection techniques, a crucial yet less studied component in this field lies in the very early stage of the pipeline, *i.e.*, the feature extraction module. Conceptually, we highlight the importance of the extracted features in that they fundamentally determine the criteria for grouping or distinguishing the adjacent frames. By analyzing the failure modes of existing state-of-the-art methods, we further argue that general-purpose global extractors, either



Figure 2. **Examples of the common failure modes of the SOTA ABD [10] method.** ABD fails to distinguish the fine discrepancy between *Crack Egg* and *Spoon Flour* as the actions occur in the far end of the camera perspective. Meanwhile, it over-segments *Fry Pancake* due to the whole-body motions of the human subject.

plain CNNs [1, 48] or learning-free descriptors [10, 21, 34] such as IDT [46], do not necessarily best fit the task of temporal action segmentation. Specifically, existing extractors pay equal attention to all the details in the video clip. Thus, the subsequent segmentation might be easily interfered with semantically characterless but numerically dramatic variations such as camera perspective shift or arbitrary large-scale movements of the human subject. In Figure 2, we provide concrete examples to shed light on our observations.

Inspired by the cognitive grounds [3, 53] that humans rely on partial components to segment complex activities, we attempt to alleviate the incompetence of global visual descriptors by exploring the merits of object-centric local features. To enhance the instantiation of this idea, we propose the framework of Object-centric Temporal Action Segmentation (OTAS), as illustrated in Figure 1. More specifically, the devised framework includes three major modules: a) a self-supervised global perception module that learns global visual features; b) a self-supervised object attention module that captures local interacting features and inter-object relational features; c) a boundary selection module that fuses the features and selects salient boundaries.

Another highlight of this paper is a discussion on the evaluation metrics, where we point out popular frame-level metrics such as Mean of Frames (MoF) are likely to be manipulated by the dominant pattern in the sequence, while the recognized acceptance thresholds of boundary-level F1 scores are far too coarse. Therefore, we appeal to the community for more thoughtful designs of metrics and in the meantime recommend a revised F1 score.

In terms of the recommended metrics, our OTAS outperforms the previous state-of-the-art approach [10] by up to **41%** on average. In terms of MoF, OTAS is superior to all the baselines on the Breakfast [17] and 50Salads [39] datasets. As supplements to the quantitative results, we conduct a user study and a qualitative case study, which cross-validate the competitive performance of our approach and the rationality of our claim on evaluation metrics. Furthermore, by directly consuming raw video inputs, OTAS is found to be efficient enough to perform real-time inference.

## 2. Related Works

### 2.1. Supervised Action Segmentation

Temporal action segmentation [7] has been thoroughly studied under fully supervised settings [6, 13, 17, 18, 42, 54], where various kinds of recurrent models are leveraged for the sequential prediction of the action labels.

As the annotation process is expensive, researchers also investigate methods that harness weaker supervision such as narrations or subtitles [2, 4, 27–29, 36, 37]. While the language guidance does provide some assistance, the temporal misalignment is inevitable since the spoken words are often ahead of the actual actions. Another line of weakly supervised works relieves the need for language guidance, where only the action types and their orders are provided [9, 19, 23, 31, 33, 47]. Furthermore, there are works relying on different weak supervision options such as an unordered set of actions [32] or timestamps [25].

### 2.2. Unsupervised Action Segmentation

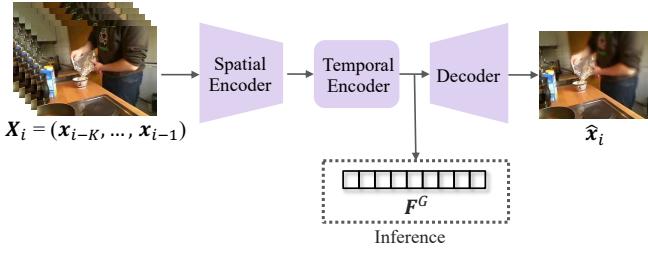
Due to expensive labor and inevitable biases brought by human annotations, unsupervised methods for action segmentation are increasingly popular. Unaware of the action types, unsupervised methods divide the video clip into neutral segments assigned with no specific classes and then depend on the Hungarian matching algorithm [20] to establish the correspondence between unlabeled predictions and ground-truth action classes. Generally, unsupervised methods can be categorized into clustering-based ones that group similar frames as well as boundary-based ones that detect salient variations.

**Clustering-based methods.** Many of the clustering-based works either assume that the videos contain the same activity [12, 21, 24, 35, 40, 45, 49, 50] or target at multi-activity collections but hold known activity labels [8, 11, 22]. A fully unsupervised pipeline is pioneered by Sener and Yao [35], who develop an iterative approach that alternates between a discriminative visual feature model and a generative temporal model. Meanwhile, clustering unknown activities is first explored by CTE [21], where continuous temporal embeddings are learned to simultaneously capture time dependencies and visual representations.

Recently, more advanced techniques such as action shuffle alternating [24] and temporally-weighted hierarchical clustering (TW-FINCH) [34] are developed to boost the segmentation performance. Nevertheless, the fact that clustering is an iterative process makes it computationally inefficient, hindering its deployment in many actual scenarios.

**Boundary-based methods.** In comparison with clustering-based counterparts that densely annotate all of the frames, boundary-based approaches are more efficient thanks to a sparse and one-pass boundary discovery process. Inspired by cognitive psychology, LSTM+AL [1] is one of the first

### (a) Global Perception Module



### (b) Object Attention Module

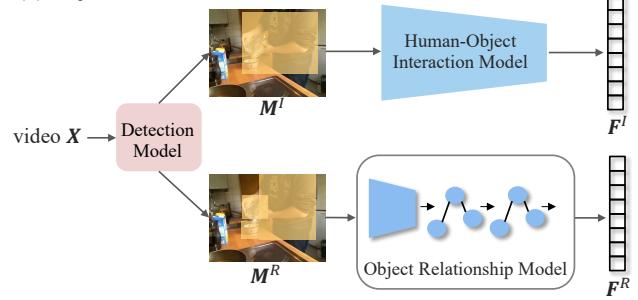


Figure 3. **The pipeline of feature extraction.** (a) **Global perception module** takes as input a sequence of  $K$  consecutive frames and outputs the *global visual features*  $\mathbf{F}^G$ . (b) **Object attention module** leverages an off-the-shelf detection model to obtain both interactive region and object masks. The interactive regions pass through a human-object interaction model to produce *local interacting features*  $\mathbf{F}^I$ . Meanwhile, the object masks are consumed by an object relationship model to generate *object relational features*  $\mathbf{F}^R$ . Both modules depend on frame prediction to validate self-supervised training.

unsupervised works that propose to detect the boundaries by analyzing the curve of errors in a self-supervised frame prediction procedure. Likewise, CoSeg [48] generate temporal features via contrastive learning and detect the boundaries by leveraging Transformer [43] models. Lately, ABD presented by Du et al. [10] proposes to detect the boundaries with learning-free offline features and apply a clustering algorithm for refinement.

Despite their remarkable performance, all of the existing methods only consider global visual features, making them vulnerable to semantically trivial but numerically predominant action variations. In contrast to the previous works, we emphasize on the feature extraction module, arguing the local interactive and relational features are also indispensable to the judgment of the ongoing actions.

### 2.3. Generic Event Boundary Detection

A closely related topic to temporal action segmentation is the newly proposed problem known as generic event boundary detection (GEBD) [15, 38, 41], where changes in perspectives, color/brightness, subjects, *etc*, are all considered as target boundaries. Unlike GEBD, we aim at the more prevailing objective of action segmentation, *i.e.*, the unsupervised detection of the semantically salient transitions from an untrimmed sequence of actions.

## 3. Method

We propose OTAS, an unsupervised framework for boundary-based temporal action segmentation that considers not only global visual features but also local object-centric features. In this section, we first present the formulation of action segmentation in Section 3.1. Then, the global perception module for global feature extraction is introduced in Section 3.2, while the object attention module for object feature extraction is introduced in Section 3.3. Fi-

nally, the resulting features are fused by the boundary selection module described in Section 3.4 for final boundaries.

### 3.1. Problem Formulation

We denote a video of unknown activity as a frame sequence  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_L)$  of length  $L$ , where  $\mathbf{x}_i$  is the  $i$ -th RGB frame. Typically,  $\mathbf{x}_i$  is in the shape of  $H \times W \times C$ , where  $H, W, C$  are the height, width, and number of channels of the frame, respectively. From the perspective of boundary-based methods, temporal action segmentation is equivalent to determining a boundary set  $\mathcal{B} = \{B_m\}_{m=1}^{M-1}$  where  $B_m \in \{1, \dots, L-1\}$ , which could split  $\mathbf{X}$  into  $M$  semantically atomic actions.

### 3.2. Global Perception Module

Endorsed by a number of previous works [1, 10, 48], global visual features could play an important role in boundary detection for temporal action segmentation. Following the practice of [1, 48], we develop a global perception module for self-supervised global feature extraction. As depicted in Figure 3(a), the spatial-temporal encoder takes as input a sequence of  $K$  consecutive past frames  $\mathbf{X}_i = (\mathbf{x}_{i-K}, \dots, \mathbf{x}_{i-1})$ , and the decoder outputs the predicted current frame  $\hat{\mathbf{x}}_i$ . To optimize the encoder-decoder architecture in an end-to-end manner, Mean Squared Error (MSE) is adopted for loss computation. The global visual feature is extracted from the backbone layer at inference.

Specifically, ResNet-50 [14] serves as the spatial encoder to transform a sequence of high-dimensional raw image inputs into the sequence of low-dimensional embeddings:

$$\mathbf{E}_i = \text{SpatEnc}(\mathbf{X}_i), \text{SpatEnc} : \mathbb{R}^{K \times H \times W \times C} \rightarrow \mathbb{R}^{K \times D_E},$$

where  $\mathbf{E}_i = (\mathbf{e}_{i-K}, \dots, \mathbf{e}_{i-1})$ , and  $\mathbf{e}_t \in \mathbb{R}^{D_E}$ .

Next, a cascading Transformer [43] architecture is leveraged to model the latent temporal relations underlying the

sequence of spatial embeddings:

$$\mathbf{F}_i^G = \text{TempEnc}(\mathbf{E}_i), \text{TempEnc} : \mathbb{R}^{K \times D_E} \rightarrow \mathbb{R}^{K \times D_E},$$

where  $\mathbf{F}_i^G = (\mathbf{f}_{i-K}^G, \dots, \mathbf{f}_{i-1}^G)$ , and  $\mathbf{f}_t^G \in \mathbb{R}^{D_E}$  is the global visual feature of the  $t$ -th frame.

To validate self-supervised training, several up-sampling and convolutional layers are stacked to decode the predicted current frame from the global visual feature of the previous frame:

$$\hat{\mathbf{x}}_i = \text{Decoder}(\mathbf{f}_{i-1}^G), \text{Decoder} : \mathbb{R}^{D_E} \rightarrow \mathbb{R}^{H \times W \times C}.$$

Once trained through the frame prediction process, the network produces distinctive and time-correlated global features from the bottleneck layer, making preparations for boundary selection to be discussed in Section 3.4. More details on the network architectures and the training process can be found in the supplementary materials.

### 3.3. Object Attention Module

The sole global features might be easily disturbed by semantically characterless but numerically dramatic variations such as arbitrary whole-body movements of the human. Therefore, as assistance to the global features, we devise an object attention module to capture the interaction among objects and their interaction with humans. As shown in Figure 3(b), the object attention module consists of an off-the-shelf detection model for pre-possessing, a human-object interaction model (Section 3.3.1) for local interacting features, and an object relationship model (Section 3.3.2) for object-relational features.

#### 3.3.1 Human-Object Interaction Model

To model the interaction between humans and objects, we rely on an out-of-domain off-the-shelf detection model [30] from Detectron2 [51] pre-trained on the COCO dataset [26] to provide a mask  $\mathbf{m}_t$  covering both the human and the interacted objects for each frame. Thus, the sequence of masked input frames  $\bar{\mathbf{X}} = (\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_L)$  that concentrate on human-object interactions can be acquired by:

$$\bar{\mathbf{X}} = \mathbf{X} \odot \mathbf{M},$$

where  $\mathbf{M} = (\mathbf{m}_1, \dots, \mathbf{m}_L)$  is the sequence of frame-wise masks and  $\odot$  refers to the element-wise product. With the masked frame sequence ready, we pass it through exactly the same encoder as that trained via the global perception module in Section 3.2. Since the outputs are supposed to contain rich interactive information, they are named as local interacting features  $\mathbf{F}^I = (\mathbf{f}_1^I, \dots, \mathbf{f}_L^I)$ ,  $\mathbf{F}^I \in \mathbb{R}^{L \times D_E}$ , which are also prepared for the boundary selection module.

#### 3.3.2 Object Relationship Model

Apart from human-object interactions, object-object relations are also helpful indicators for action segmentation. OTAS proposes to model the relationships of the detected object patches using graphs, as shown in Figure 4. More specifically, an object relation look-up table [52] is utilized to construct a relational graph for each frame, connecting semantically reactive and spatially neighboring objects. Afterward, a Graph Neural Network (GNN) [5] is trained by predicting the future frames from both the image features and the cropped object features. During inference time, latent features of the GNN are extracted, known as the object-relational features.

**Object relation look-up table.** We aim to extract relations among objects that are either similar in semantics or possible for mutual interactions. Hence, we build an object relation look-up table by collecting the relations from the Visual Genome dataset [16] which concern objects concurrently appearing in the COCO dataset.

**Graph construction.** We construct a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  of object relations for each frame. The nodes  $\mathcal{V} = \{v^{(1)}, \dots, v^{(N)}\}$  correspond to the  $N$  detected objects within the frame. The edge  $\{v^{(p)}, v^{(q)}\}$  between node  $v^{(p)}$  and node  $v^{(q)}$  exists if and only if a)  $v^{(p)}$  and  $v^{(q)}$  are semantically related according to the look-up table; and b) the distance between the two masks of the objects is within a threshold  $\theta_r$ . In Figure 4(a), we illustrate how to construct the graph with the help of the object relation look-up table.

**Graph Neural Network.** To extract the latent inter-object relationship information embedded in the object graph, we train the network by forcing it to predict the current frame  $\hat{\mathbf{x}}_i$  from the information contained in the past frame  $\mathbf{x}_{i-1}$ . A visual illustration is shown in Figure 4(b).

For each node  $v_i^{(p)} \in \mathcal{V}_i$ , its initial node representation  $\mathbf{r}_i^{(p)} \in \mathbb{R}^{H \times W \times 2C}$  is the concatenation of the whole image frame and the masked patch of the  $p$ -th object:

$$\mathbf{r}_i^{(p)} = \mathbf{x}_i \parallel (\mathbf{x}_i \odot \mathbf{m}_i^{(p)}),$$

where  $\mathbf{m}_i^{(p)}$  is the mask of the  $p$ -th detected object in the  $i$ -th frame, and  $\parallel$  denotes the concatenation operation.

Our detailed implementation of the Graph Neural Network is adapted from the dynamic graph attention variant of Graph Attention Networks (GATs) [5, 44]. Specifically, a self-attention mechanism for node  $v_i^{(p)}$  is used to attend over its neighbors  $\mathcal{N}_i^{(p)} = \{v_i^{(q)} \in \mathcal{V}_i \mid \{v_i^{(p)}, v_i^{(q)}\} \in \mathcal{E}_i\}$ , where a scoring function  $s$  computes the attention score of a node  $v_i^{(p)}$  and its neighbor  $v_i^{(q)}$ :

$$s(\mathbf{r}_i^{(p)}, \mathbf{r}_i^{(q)}) = \mathbf{a}^\top \text{LeakyReLU} \left( \mathbf{W} \cdot (\mathbf{r}_i^{(p)} \parallel \mathbf{r}_i^{(q)}) \right),$$

where  $\mathbf{a} \in \mathbb{R}^{2D_E}$  and  $\mathbf{W} \in \mathbb{R}^{2D_E \times 2D_E}$  are learnable parameters. Then, the scoring function is leveraged to calcu-

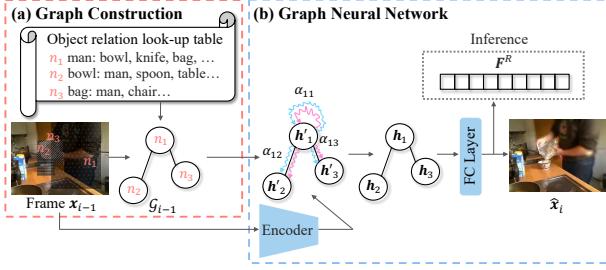


Figure 4. **Object relationship model.** (a) **Graph construction.** The graph is built upon the object relation look-up table while the object masks are generated from the object detection model. (b) **Graph Neural Network.** The object masks are consumed by an encoder to initialize each node’s representation. Then, multi-head self-attention is used to process the graph information. Finally, a fully connected layer is applied to compute the object relational features. Here, we omit the decoder at training time for clarity.

late the hidden node representation  $\mathbf{h}_i^{(p)} \in \mathbb{R}^{2D_E}$ :

$$\alpha_i^{pq} = \frac{\exp\left(s\left(\mathbf{r}_i^{(p)}, \mathbf{r}_i^{(q)}\right)\right)}{\sum_{v_i^{(q')} \in \mathcal{N}_i^{(p)}} \exp\left(s\left(\mathbf{r}_i^{(p)}, \mathbf{r}_i^{(q')}\right)\right)},$$

$$\mathbf{h}_i^{(p)} = \text{ReLU}\left(\sum_{v_i^{(q)} \in \mathcal{N}_i^{(p)}} \alpha_i^{pq} \mathbf{W} \cdot \mathbf{r}_i^{(q)}\right).$$

The entire hidden representation  $\mathbf{H}_i \in \mathbb{R}^{N \times 2D_E}$  of a frame is the concatenation of the hidden representations for all the nodes in  $\mathcal{V}_i$ . Finally, an additional fully-connected layer consumes  $\mathbf{H}_i$  and outputs the final object relational feature  $\mathbf{F}_i^R \in \mathbb{R}^{D_E}$ . Besides, at training time,  $\mathbf{F}_i^R$  is further fed to a decoder similar to that in Section 3.2 for frame prediction so as to enable the self-supervised training process.

### 3.4. Boundary Selection Module

Given the access to the global visual features  $\mathbf{F}^G$ , the local interacting features  $\mathbf{F}^I$ , and the object-relational features  $\mathbf{F}^R$ , the boundary selection module is desired to effectively integrate the three features and sensitively capture the salient variations. As shown in Figure 5, we select the final boundaries from the local maximums of the sequence of temporal feature differences.

For an arbitrary feature  $\mathbf{f}_i^X$  of the frame  $x_i$  taken from the set  $\{\mathbf{f}_i^G, \mathbf{f}_i^I, \mathbf{f}_i^R\}$ , we define its temporal feature difference  $\epsilon_i^X$  as the accumulated squared- $\ell_2$  distance between a sequence of  $K$  consecutive features prior to  $x_i$ , *i.e.*,  $(\mathbf{f}_{i-K}^X, \dots, \mathbf{f}_{i-1}^X)$ , and another sequence of  $K$  consecutive frames starting from  $x_i$ , *i.e.*,  $(\mathbf{f}_i^X, \dots, \mathbf{f}_{i+K-1}^X)$ . Then, a frame is considered as a boundary candidate if its temporal feature difference is the local maximum within an interval  $\alpha$ . We obtain three sets of boundary candidates from the sequence of global feature differences  $\epsilon^G$ , interacting feature differences  $\epsilon^I$ , and object-relational feature differences  $\epsilon^R$ .

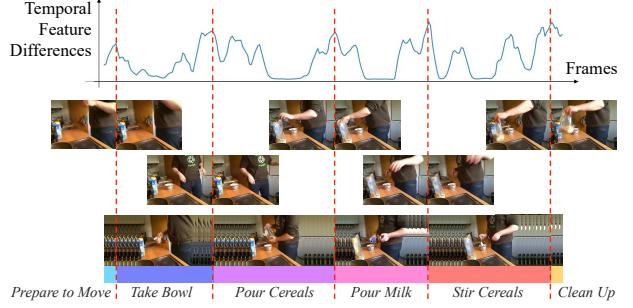


Figure 5. **Boundary selection.** Local maximums of the temporal feature differences are regraded as the boundary candidates, who vote to decide the final boundary predictions.

Now with the candidates at hand, we design a specialized voting mechanism to determine the final boundary predictions. In broad strokes, we accept two types of boundaries: a) a boundary that is agreed by all of the three candidate sets; or b) a boundary whose temporal feature difference is significantly salient. To instantiate this idea, we calculate a weighted confidence score for all the candidate boundaries:

$$S = \beta_G \epsilon^G + \beta_I \epsilon^I + \beta_R \epsilon^R,$$

where  $\beta_G, \beta_I, \beta_R$  are tunable. To obtain the boundaries that are unanimously agreed by all the sets, we examine each candidate generated from the global features. If it has neighboring candidates from both other sets within a small time range  $\theta_n$ , we call these neighbors constituting a boundary cluster. Within each boundary cluster, we select the one with the highest confidence score  $S$  as the final prediction. For those candidates that are significantly salient in one specific feature but do not have neighbors from other sets, we accept those enjoying the confidence scores that are twice larger than the maximum of the previously selected ones.

## 4. Experiments

We first introduce the experimental setups in Section 4.1. Next, we call for the community’s attention to more reasonable metrics for unsupervised temporal action segmentation in Section 4.2. Then, we compare the proposed OTAS with competitive baselines in Section 4.3 in terms of quantitative and qualitative performance, user study, and computational cost. Lastly, we ablate the important factors that contribute to the effectiveness of OTAS in Section 4.4.

### 4.1. Setup

#### 4.1.1 Datasets

**Breakfast** [17] includes 1,712 videos comprising 10 breakfast cooking activities. The duration of the videos varies dramatically from 30 seconds to 7 minutes. Furthermore, the videos may contain occlusions and different viewpoints.

**50Salads** [39] includes altogether 4 hours videos of mixing salad, whose average length is 5 minutes. It defines 17 *mid-level* activities and 9 *eval-level* activities.

**INRIA** [2] includes 150 videos of complex activities not restricted to cooking with an average length of 2 minutes. Specially, it carries up to 83% ratio of background frames that are extremely diverse in terms of visual appearance.

#### 4.1.2 Hyper-Parameters

We down-sample the rates of all the videos to 5 fps and resize each frame to  $256 \times 256 \times 3$ . For hyper-parameters, we set  $D_E = 2048$  as the feature dimension,  $K = 5$  frames which lasts for one second as the sequence length for frame prediction and boundary selection,  $\theta_r = 80$  pixels for the distance threshold between two object masks in graph construction, and  $\alpha = 15$  frames,  $\beta_G = \beta_I = 1$ ,  $\beta_R = 0.3$ ,  $\theta_n = 2$  seconds for the boundary selection module.

More implementation details can be found in the supplementary materials.

#### 4.1.3 Baselines

OTAS is compared with a series of competitive baselines including the clustering-based CTE [21] and TW-FINCH [34], and the boundary-based LSTM+AL [1], Coseg [48] and ABD [10]. Due to the demand of clustering methods for a pre-set clusters numbers, we reveal the average number of actions to CTE [21], TW-FINCH [34], and ABD [10]. Additionally, we also evaluate the naive setting of *Equal Split*, which equally divides the video into the same number of segments.

### 4.2. Evaluation Metrics

Owing to the dramatically varying patterns and lengths of the videos as well as the relatively subjective ground-truth annotations for temporal action segmentation, we find current metrics are sometimes too crude to rely on. As the community always pursues a fair and rigorous evaluation, we think it necessary to dive into the details of the metrics.

**Frame-level metrics.** Dense frame-level scores such as Mean of Frames (MoF) or Intersection over Unions (IoU) are the most commonly used metrics for supervised temporal action segmentation [6, 17, 31], because they accurately measure the distance between the predictions and the ground-truth annotations. In unsupervised settings, however, they become slightly unnatural since the unsupervised algorithms produce untagged neutral segments and we have to rely on the Hungarian algorithm [20] to artificially build the correspondence between predictions and annotations, which may cause additional biases. Moreover, frame-level metrics are likely to be manipulated by the dominant pattern in the sequence, as illustrated in Figure 6. This bias can be cross-verified by the quantitative results in Table 1,



Figure 6. **Example of unreasonable MoF score.** We compare the ground-truth annotations on the clip of *P47\_webcam01\_salad* in Breakfast [17] and the results generated by ABD [10]. While ABD fails to detect more than 80% of the boundaries, it reaches a high MoF of 56.78, because the MoF is manipulated by the dominant frames standing for *Cut Fruit* marked in purple.

where even the naive setting of *Equal Split* could achieve relatively decent performance in MoF.

**Boundary-level metrics.** Compared with the potentially biased frame-level metrics, we prefer the boundary-level F1 score, which directly measures the discrepancy between the predicted and the annotated boundaries. Nevertheless, the side effect is that we have to manually decide the threshold within which a predicted boundary is accepted as a positive one. To our surprise, the conventional procedure [38, 48] is to take 5% of the video duration as the distance threshold (*i.e.*, 15 seconds for a five-minute video), which we believe is too coarse to distinguish right from wrong.

Thus, we advocate evaluation with F1(*small*) which has a smaller fixed threshold of 2 seconds. With respect to conventions, we also report the F1(*large*) scores following the traditional 5% threshold, but they are marked in gray for distinction.

**Human evaluation.** Though very expensive, the probably most unbiased way of evaluation is always to invite humans to make the judgment, especially when the ground-truth annotations themselves are more or less subjective. Since personal characteristics such as race, gender, *etc.*, are less likely to impact decisions on action segmentation, we recruit 33 anonymous volunteers from the Internet. Each of them is provided with 20 video clips randomly picked from the Breakfast [17] dataset with a total length of 40.3 minutes, accompanied with five segmentation results per video in a shuffled order — one copied from ground-truth, one generated by OTAS, and the other three generated by previous works [10, 21, 34]. Given no instructions or clues on how to segment (*i.e.*, no granularity reference), the volunteers are asked to rank the five options. Afterward,  $(6 - \text{rank})$  is considered as the ranking score for the video (*e.g.*, ranking No.2 gets 4 points), and the averaged ranking score is used for evaluation. The average completion time per person is two and a half hours.

### 4.3. Experimental Results

#### 4.3.1 Quantitative Results

The quantitative results of OTAS against the baselines in terms of frame-level MoF and boundary-level F1 scores on the three datasets are shown in Table 1, 2, 3, respectively. F1(*small*) is recommended over MoF because it is in better

	MoF	F1( <i>large</i> )	F1( <i>small</i> )
Equal Split	54.06	33.08	14.04
LSTM+AL[1]	42.90	—	—
CTE[21]	60.50	35.60	19.52
TW-FINCH[34]	62.70	43.35	23.82
CoSeg[48]	53.10	54.70	—
ABD[10]	64.00	49.56	27.93
<b>OTAS (ours)</b>	<b>67.90</b>	<b>62.13</b>	<b>44.49</b>

Table 1. **Quantitative results on Breakfast** [17]. OTAS significantly outperforms the second best by 59% in terms of F1(*small*), and leads the chart in terms of MoF as well.

	MoF	F1( <i>large</i> )	F1( <i>small</i> )
<i>Eval-level</i>			
Equal Split	47.40	63.97	15.92
CTE[21]	53.92	41.82	11.80
LSTM+AL[1]	60.60	—	—
CoSeg[48]	64.10	71.80	—
TW-FINCH[34]	71.10	47.15	22.72
ABD[10]	71.40	66.02	30.50
<b>OTAS (ours)</b>	<b>73.57</b>	<b>72.72</b>	<b>49.58</b>
<i>Mid-level</i>			
Equal Split	33.10	<b>78.35</b>	22.88
CTE[21]	43.68	41.93	11.76
TW-FINCH[34]	66.50	70.19	19.18
ABD[10]	71.80	70.80	40.47
<b>OTAS (ours)</b>	<b>72.42</b>	71.07	<b>53.13</b>

Table 2. **Quantitative results on 50Salads** [39]. OTAS significantly outperforms the second best by 63% in terms of F1(*small*) on *Eval-level* and by 31% on *Mid-level*. OTAS also leads the chart in terms of MoF. Notably, Equal Split should beat all the other fancy techniques in terms of the unreasonable F1(*large*) metrics on *Mid-level*. We attribute the phenomenon to the coincidence that each action of mixing salads takes approximately the same time.

accord with the requirements for unsupervised action segmentation. F1(*large*) is also presented but not worth noticing due to its false thresholds.

Equipped with local object attention, OTAS significantly surpasses the previous state-of-the-art method [10] by 41% on average in terms of F1(*small*), and outperforms all the baselines in terms of MoF on Breakfast and 50Salads.

Note that F1(*small*) reports a performance ranking generally in line with MoF but could better distinguish the competence of different approaches, while F1(*large*) reports a ranking that is rather inconsistent with its counterparts. These observations could cross-validate our claim on the evaluation metrics.

### 4.3.2 User Study

As a helpful complement to quantitative metrics, the user study results are listed in Table 4. Humans tend to rate OTAS over the baselines. They even prefer OTAS to the

	MoF	F1( <i>large</i> )	F1( <i>small</i> )
Equal Split	30.2	57.54	24.58
CTE[21]	39.08	<b>70.27</b>	27.62
CoSeg[48]	47.90	53.70	—
TW-FINCH[34]	56.70	58.10	24.27
ABD[10]	<b>67.20</b>	64.92	34.18
<b>OTAS (ours)</b>	65.71	66.53	<b>37.28</b>

Table 3. **Quantitative results on INRIA** [2]. OTAS outperforms the second best by 9% in terms of F1(*small*). It achieves comparable performance to ABD [10] in terms of MoF. We attribute the degradational performance of OTAS to up to 83% ratio of background frames on the INRIA dataset.

	Avg. Ranking Score
CTE[21]	2.5025
TW-FINCH[34]	2.722
ABD[10]	2.933
Ground-Truth	3.167
<b>OTAS (ours)</b>	<b>3.331</b>

Table 4. **User study results on Breakfast**. Humans tend to rate OTAS over all the baselines. Surprisingly, OTAS is even considered to outperform the ground-truth annotations in the dataset.

ground-truth annotations, indicating that human-annotated labels are sometimes counter-intuitive. By carefully examining the dataset, we discover the ground-truth labels are deteriorated by inconsistent segmentation granularity and negligence on fine object details, as illustrated in Figure 8.

Besides, note that the ranking score is largely in line with the F1(*small*) metrics in Table 1, 2, 3, which once again cross-validates our preference to evaluation metrics.

### 4.3.3 Qualitative Results

The qualitative case study on a challenging video clip with dramatically varying segment lengths is shown in Figure 7. Thanks to the collaboration between the global and local features, OTAS not only successfully captures the detailed variations easily omitted between consecutive short actions, but also effectively alleviates the over-segmentation problem commonly encountered in a continuous long segment.

### 4.3.4 Computational cost

The computational cost is compared in Table 5. Following ABD [10], we report the inference time to segment 2000 frames with the visual features already prepared. As a boundary-based method, OTAS is as efficient as ABD while evidently more efficient than clustering-based methods.

Additionally, we report the often ignored cost of feature extraction and the more practical overall efficiency in fps, which involves both feature extraction and segmentation inference. Interestingly, we find that the overall expenses are

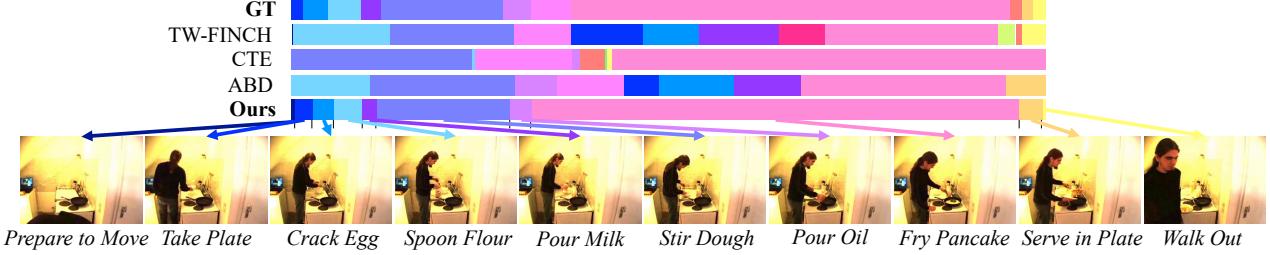


Figure 7. **Case study of the segmentation results of ground-truth (GT), TW-FINCH [34], CTE [21], ABD [10], and Ours.** OTAS successfully recognizes consecutive fine actions at the beginning while avoids to over-segment the long action of *Fry Pancake*.

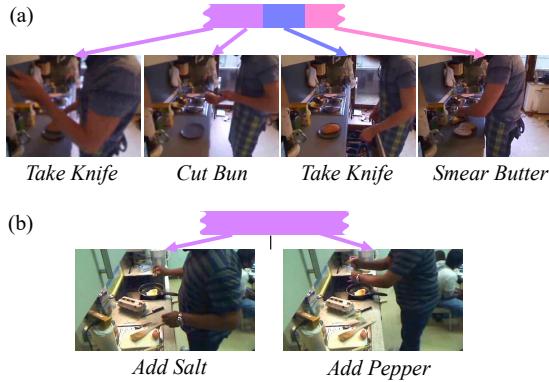


Figure 8. **Examples of common flaws in ground-truth (GT).** (a) **Inconsistent granularity.** GT distinguishes *Take Knife* and *Smear Butter* but does not distinguish *Take Knife* and *Cut Bun*. (b) **Confusion with similar actions.** GT perceives *Add Salt* and *Add Pepper* as one action while users tend to separate them.

	Feature (s)	Inference (s)	Overall (fps)
CTE [21]	229.1	217.94	4.5
TW-FINCH [34]	229.1	0.16	8.7
ABD [10]	229.1	<b>0.02</b>	8.7
OTAS (ours)	<b>5.58</b>	<b>0.02</b>	<b>357.1</b>

Table 5. **Comparison of the computational cost.** Boundary-based methods (OTAS and ABD [10]) are significantly more efficient than clustering-based methods (CTE [21] and TW-FINCH [34]) when inference, while the overall computational cost is bounded by the feature extraction process. OTAS obtains overall efficiency of 357.1 fps on a single NVIDIA GeForce RTX 2080 Ti, indicating its potential for real-time applications.

bounded by the feature preparation procedure. By explicitly consuming raw video inputs, OTAS avoids the time-consuming computation of IDT [46] features leveraged by the baselines, making it stands out in overall efficiency.

#### 4.4. Ablation study

The ablation study in Table 6 demonstrates the effectiveness of the local object attention module. A more intuitive

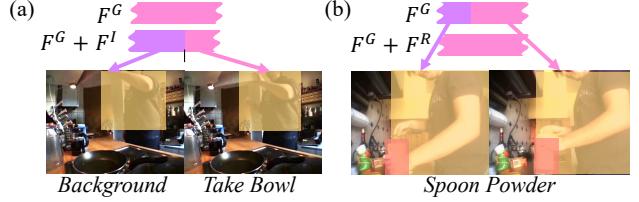


Figure 9. **Effectiveness of local features.** (a) The action of *Take Bowl* occurs in a small interacting region.  $F^I$  successfully detects it while  $F^G$  alone fails. (b) *Spoon Powder* is a large-scale action in the near end of the camera perspective, which leads to significant variations in the global feature.  $F^R$  avoids over-segmentation with the discovery of invariant object relations.

	F1 <small>(small)</small>	MoF
$F^G$	37.46	65.99
$F^I$	36.54	65.19
$F^R$	37.29	61.33
$F^G + F^I$	43.09	65.12
$F^G + F^R$	44.11	67.85
$F^G + F^I + F^R$	<b>44.49</b>	<b>67.90</b>

Table 6. **Ablation study on Breakfast.** The ensemble of global and local features outperforms any sole feature. The ensemble of all three features leads to the best performance.

comparison between the qualitative behaviors of different feature combinations is shown in Figure 9. More ablation studies can be found in the supplementary materials.

## 5. Conclusion

In this paper, we take local visual features into consideration to tackle the problem unsupervised temporal action segmentation. We discuss the rationality of the evaluation metrics, and more importantly, propose the framework of OTAS that combines global visual features, local interacting features, and object relational features. OTAS achieves state-of-the-art performance on quantitative metrics, user studies, and computational cost.

## 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 (CVPR)*, June 2019. 1, 2, 3, 6, 7

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 1, 2, 6, 7

[3] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 2

[4] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *Proceedings of the IEEE international conference on computer vision*, pages 4462–4470, 2015. 2

[5] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 4

[6] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan Al-Regib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9454–9463, 2020. 1, 2, 6

[7] Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern technique. *arXiv preprint arXiv:2210.10352*, 2022. 1, 2

[8] Guodong Ding and Angela Yao. Temporal action segmentation with high-level complex activity labels. *arXiv preprint arXiv:2108.06706*, 2021. 2

[9] 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

[10] 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. 1, 2, 3, 6, 7, 8

[11] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. In *European Conference on Computer Vision*, pages 557–573. Springer, 2020. 2

[12] Ehsan Elhamifar and Zwe Naing. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6341–6350, 2019. 2

[13] Yazan Abu Farha and Jurgen Gall. Ms-ten: 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

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[15] Hyolim Kang, Jinwoo Kim, Taehyun Kim, and Seon Joo Kim. Uboco: Unsupervised boundary contrastive learning for generic event boundary detection. *arXiv preprint arXiv:2111.14799*, 2021. 3

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 4

[17] 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. 1, 2, 5, 6, 7

[18] 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

[19] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 2

[20] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2, 6

[21] 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. 1, 2, 6, 7, 8

[22] Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. Unsupervised activity segmentation by joint representation learning and online clustering. *arXiv preprint arXiv:2105.13353*, 2021. 2

[23] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6243–6251, 2019. 2

[24] Jun Li and Sinisa Todorovic. Action shuffle alternating learning for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12628–12636, 2021. 1, 2

[25] 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. 2

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[27] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What’s cookin’? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558*, 2015. 2

[28] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 9879–9889, 2020.

[29] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. [2](#)

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. [4](#)

[31] 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. [1, 2, 6](#)

[32] 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](#)

[33] 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](#)

[34] 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. [1, 2, 6, 7, 8](#)

[35] 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*, pages 8368–8376, 2018. [2](#)

[36] 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](#)

[37] Yuhang Shen, Lu Wang, and Ehsan Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2021. [2](#)

[38] Mike Zheng Shou, Stan Weixian Lei, Weiyao Wang, Deepthi Ghadiyaram, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8075–8084, 2021. [3, 6](#)

[39] 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. [1, 2, 6, 7](#)

[40] Sirnam Swetha, Hilde Kuehne, Yogesh S Rawat, and Mubarak Shah. Unsupervised discriminative embedding for sub-action learning in complex activities. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2588–2592. IEEE, 2021. [2](#)

[41] Jiaqi Tang, Zhaoyang Liu, Chen Qian, Wayne Wu, and Limin Wang. Progressive attention on multi-level dense difference maps for generic event boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3355–3364, 2022. [3](#)

[42] Yansong Tang, Jiwen Lu, and Jie Zhou. Comprehensive instructional video analysis: The coin dataset and performance evaluation. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3138–3153, 2020. [2](#)

[43] 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. [3](#)

[44] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017. [4](#)

[45] 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](#)

[46] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. [2, 8](#)

[47] Lin Wang, Xingfu Wang, Ammar Hawbani, and Yan Xiong. End to end alignment learning of instructional videos with spatiotemporal hybrid encoding and decoding space reduction. *Applied Sciences*, 11(11):4954, 2021. [2](#)

[48] Xiao Wang, Jingen Liu, Tao Mei, and Jiebo Luo. Coseg: Cognitively inspired unsupervised generic event segmentation. *arXiv preprint arXiv:2109.15170*, 2021. [1, 2, 3, 6, 7](#)

[49] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Unsupervised action segmentation with self-supervised feature learning and co-occurrence parsing. *arXiv e-prints*, pages arXiv-2105, 2021. [2](#)

[50] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1819–1828, 2022. [2](#)

[51] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. [4](#)

[52] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018. [4](#)

[53] Jeffrey M Zacks and Khena M Swallow. Event segmentation. *Current directions in psychological science*, 16(2):80–84, 2007. [2](#)

[54] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. [2](#)