CoSeg: Cognitively Inspired Unsupervised Generic Event Segmentation

Xiao Wang, Graduate Student Member, IEEE, Jingen Liu, Senior Member, IEEE, Tao Mei, Fellow, IEEE, and Jiebo Luo, Fellow, IEEE

Abstract—Some cognitive research has discovered that humans accomplish event segmentation as a side effect of event anticipation. Inspired by this discovery, we propose a simple yet effective end-to-end self-supervised learning framework for event segmentation/boundary detection. Unlike the mainstream clustering-based methods, our framework exploits a transformer-based feature reconstruction scheme to detect event boundaries by reconstruction errors. This is consistent with the fact that humans spot new events by leveraging the deviation between their prediction and what is perceived. Thanks to their heterogeneity in semantics, the frames at boundaries are difficult to be reconstructed (generally with large reconstruction errors), which is favorable for event boundary detection. In addition, since the reconstruction occurs on the semantic feature level instead of the pixel level, we develop a temporal contrastive feature embedding (TCFE) module to learn the semantic visual representation for frame feature reconstruction (FFR). This procedure is like humans building up experiences with “long-term memory.” The goal of our work is to segment generic events rather than localize some specific ones. We focus on achieving accurate event boundaries. As a result, we adopt the F1 score (Precision/Recall) as our primary evaluation metric for achieving accurate event boundaries. As a result, we adopt the F1 score (Precision/Recall) as our primary evaluation metric for achieving accurate event boundaries. As a result, we adopt the F1 score (Precision/Recall) as our primary evaluation metric for achieving accurate event boundaries.

Index Terms—Generic event boundary detection (GEBD), self-supervised learning, transformer, video event segmentation.

I. INTRODUCTION

TEMPORAL event segmentation in untrimmed long videos is crucial to complex activity understanding and has received increasing attention recently. Here, an event is defined as “a meaningful video segment of time at a given location that an observer conceives to have a beginning and an end” [1], and a complex activity generally consists of a sequence of such events.

Manuscript received 14 September 2021; revised 28 October 2022; accepted 27 March 2023. Date of publication 4 May 2023; date of current version 4 September 2024. (Corresponding author: Jingen Liu.)

Xiao Wang is with the Department of Computer Science, Purdue University, West Lafayette, IN 47906 USA.
Jingen Liu is with JD AI Research, Mountain View, CA 94039 USA (e-mail: jingen.liu@gmail.com).
Tao Mei is with JD AI Research, Beijing 100101, China.
Jiebo Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA.
Digital Object Identifier 10.1109/TNNLS.2023.3263387

Fig. 1. Motivation of our CoSeg event segmentation. (a) To imitate human’s long-term memory, a self-supervised feature representation network is trained to distinguish intravideo frames from inter-video frames in the video dataset. In other words, this feature encoding aims at learning a distinguish feature representation globally. (b) FFR network is trained to mimic the online memory to capture local temporal relationships within a local snippet context. At this stage, it aims to learn a reconstruction model that can reconstruct the masked frames given a video snippet.

To automatically segment one activity in a video into different event segments, the supervised approaches [2], [3], [4], [5], [6] are trained with dense frame-level annotations. Although they have achieved good performance, the expensive and time-consuming annotation is a hindrance to large-scale applications. In contrast, weakly supervised methods [2], [7], [8], [9], [10] attempt to acquire supervision signals from the video’s side information, including subtitles, text scripts, and audios. However, such side information may not be reliable for training due to the lack of precise alignment with the video in a temporal order.

Therefore, unsupervised event segmentation is receiving increasing attention. Based on the observation that different frames of the same event usually share similar visual features, most recent unsupervised approaches [5], [11], [12], [13], [14] exploit various clustering mechanisms to uncover the subactions/events in the dataset and then apply the clusters for event segmentation. However, the clustering procedure disregards important and meaningful temporal relationships between video segments. Besides, it is difficult to determine the number of clusters when conducting clustering on video datasets, especially for an uncurated one. As a result, the clustering-based approaches are not good at dealing with...
generic event segmentation for videos “in the wild.” Essentially, the clustering mechanism is not Well-aligned with humans’ perception system because humans perceive an event boundary without explicitly grouping subactions.

In fact, some recent cognitive studies [15], [16], [17] have illustrated humans segment events automatically and subconsciously. According to the event segmentation theory (EST) [16], the human perceptual system spontaneously segments an activity into different events as a side effect of anticipating upcoming events. In other words, a new event is triggered by the human perceptual system due to the failure of event anticipation or reconstruction. To further understand this phenomenon, additional cognitive experiments [17] have been conducted to demonstrate that the event anticipation procedure is closely related to the working memory system, which includes the long-term memory linked to previously learned knowledge and the online memory associated with the current activity. Humans exploit the working memory to anticipate the upcoming stream and its failure to identify the event boundary. Therefore, we propose to design neural-network-based modules to mimic the long-term and online memory in the human perceptual system.

Inspired by these cognitive studies, in this article, we propose an end-to-end self-supervised generic event segmentation framework called CoSeg. As shown in Fig. 1, we devise a self-supervised feature representation network to imitate human’s long-term memory and a transformer-based frame feature reconstruction (FFR) scheme to mimic online learning. It effectively exploits the temporal relationships of video frames to generate an arbitrary frame in its feature space. We observed that the frames around an event boundary are usually semantically heterogeneous. As a result, it is difficult to reconstruct those boundary frames, and such reconstructions generally carry higher reconstruction errors, which can be exploited to detect the boundary. This phenomenon also agrees with the EST that humans detect new events due to the failure of event reconstruction.

Our frame reconstruction occurs in the feature space instead of the pixel space because our system aims at “understanding” a frame to tell whether it belongs to the current event. To better support the reconstruction in feature space, we develop a self-supervised feature representation learning network, which uses a contrastive learning mechanism to distinguish whether two frames are from a specific local context/event. Leveraging the carefully designed temporal positive and negative training pairs, our temporal contrastive feature embedding (TCFE) learns a solid semantic feature representation for frame reconstruction. This feature representation learning acts like a mechanism when humans build up their long-term memory. Unlike the mainstream clustering-based unsupervised methods, our framework opens a new avenue for unsupervised generic event segmentation. It can be trained with any non-curated dataset (no predefined action is needed) because our approach does not rely on the discovery of explicit common patterns (i.e., action categories or pseudolabels). As shown in our following experiments, a model trained on a different dataset can still achieve comparable results, proving our approach’s transferability. Thanks to contrastive feature embedding, our model is more generalized to avoid the oversegmentation problem. In addition, our reconstruction can be conducted within a short range of frames (i.e., local context), which acts like humans’ “online memory.”

In contrast to some conventional works in event segmentation and action detection, which focus on localizing predefined action classes [11], [12], [13], [14], our CoSeg system aims at detecting generic and taxonomy-free event boundaries without predefined action categories. Specific action localization is out of the scope of our work. Our work can be categorized into the generic event boundary detection (GEBD) [18], which detects generic event boundaries such that a video is divided into natural, meaningful segments. Compared with class-specific event segmentation, the GEBD task is more general and applicable to train and evaluate long-frame datasets, such as TAPOS [19] and Kinetics [20]. It is valuable to various applications including video summarization and video-level classification [18].

In short, our contributions can be summarized as follows.

1) To the best of our knowledge, we are the first to leverage cognitive studies on how humans segment events to design a simple yet effective end-to-end self-supervised framework for generic event segmentation.

2) Unlike previous clustering-based approaches, our CoSeg system can work on non-curated datasets. It is able to conduct generic event segmentation, not only limited to predefined events. Hence, it can scale to any generic video.

3) Compared with previous methods, our CoSeg has demonstrated strong model transferability across different datasets and better generalization. As a result, it can alleviate the oversegmentation issue and the problem of overfitting to dominant classes.

4) Our CoSeg has been extensively evaluated on four widely used benchmark datasets: Kinetics-GEBD [18], Breakfast [3], 50 Salads [7], and National Institute for Research in Digital Science and Technology (INRIA) Instructional [21]. All the experiments have demonstrated that our CoSeg can outperform previous approaches by a large margin in various metrics.

II. RELATED WORK

A. Supervised Event Segmentation

The fully supervised approaches leverage densely annotated frame-level annotations to train classifiers and assign event labels to each frame. The event segmentation is conducted by temporally grouping the frames with the same labels into some coherent “chunks.” The representative framewise labeling approaches include support vector machine (SVM) [3], hidden Markov models (HMMs) [3], SPATIO-temporal convolutional neural networks (SCNs) [5], and temporal convolutional neural networks (TCNs) [22].

B. Weakly Supervised Event Segmentation

To alleviate the need for a large number of annotations, researchers have explored the weakly supervised methods,
which attempt to train a model with less supervision. Some works [2], [7], [8], [9], [10] propose to leverage the video’s additional modalities, such as subtitles, text scripts, and audios, to provide supervision signal for model training. One of their assumptions is acceptable alignment between visual frames and other modalities. Unfortunately, the quality of the alignment is not guaranteed at most time. To overcome this limitation, some works [2], [6], [23], [24], [25] try to use the temporal order of actions in a video without alignment for weak supervision.

C. Unsupervised Event Segmentation

Video segment clustering is the most widely used unsupervised mechanism by previous work [11], [12], [13], [14]. The basic idea is grouping frames into subactions or events, which results in event segmentation. Leveraging the advanced frame representation by the pretrained network on ImageNet, researchers have proposed various optimization methods to achieve better temporal clusters, such as Frank–Wolfe algorithm [26], generalized mallows model (GMM) [13], Viterbi algorithm [12], and U-Net decoding [14]. Although the clustering procedure is similar to the aforementioned long-term memory, the clustering results (i.e., segmentation) are susceptible to the number of clusters, which is hard to determine. In addition, clustering ignores the temporal order of events, which is an essential clue for event segmentation. Therefore, to achieve better performance of those methods, they typically require adjusting the number of clusters according to the prior knowledge of the dataset. In contrast, no such prior information is needed for our CoSeg, because it is able to detect an event boundary once it fails to reconstruct the boundary frames. CoSeg does not know the event category, as it treats each event in one video as a new category. CoSeg only concentrates on the difference between the consecutive events, while ignoring the relative relationship of different events from the same video.

Cognitive research [1], [15], [16], [17] shows that human segmenting events is a side effect of human anticipation. Thus, Aarkur and Sarkar [27] propose an RNN (i.e., LSTM) framework to predict the incoming frame using the past information and use the prediction errors to segment events. However, we argue that people generally rely on not only information and use the prediction errors to segment events. Therefore, we formulate event segmentation as a reconstruction problem rather than a predictive notion, as it treats each event in one video as a new category. CoSeg does not know the event category, as it treats each event in one video as a new category. CoSeg only concentrates on the difference between the consecutive events, while ignoring the relative relationship of different events from the same video.

Cognitive research [1], [15], [16], [17] shows that human segmenting events is a side effect of human anticipation. Thus, Aarkur and Sarkar [27] propose an RNN (i.e., LSTM) framework to predict the incoming frame using the past information and use the prediction errors to segment events. However, we argue that people generally rely on not only information and use the prediction errors to segment events. Therefore, we formulate event segmentation as a reconstruction problem rather than a predictive notion, as it treats each event in one video as a new category. CoSeg does not know the event category, as it treats each event in one video as a new category. CoSeg only concentrates on the difference between the consecutive events, while ignoring the relative relationship of different events from the same video.

III. PROPOSED END-TO-END METHOD

Given a generic video, the event segmentation task aims at detecting the boundary (frames) of two events. To this end, most unsupervised event segmentation approaches usually carry out frame clustering over a curated video dataset to label every frame with a pseudoevent label (i.e., cluster ID). However, clustering the high-dimensional frame features is very challenging, and its performance is sensitive to the number of clusters that is unknown for non-curated data. In fact, we humans segment events by detecting the deviation between what we are anticipating/expecting and perceiving. Inspired by this, in this section, we present a novel unsupervised event segmentation framework, which leverages the FFR errors to spot the event boundary.

Fig. 2 shows the overview of our end-to-end unsupervised learning system (CoSeg), which consists of two major modules: contrastive temporal feature embedding (CTFE) and FFR. Given a video segment/snippet $s$, which is a sequence of $T$ frames, we first apply feature extractor $f_0$ to obtain $T$ features $h_t (t \in [0, T − 1])$. During training, the CTFE module is framed as self-supervised contrastive feature learning, taking intra-segment frames as positive and inter-segment frames as negative pairs. The goal is to learn a distinctive semantic representation to distinguish event frames, acting like humans’ “long-term memory.” Meanwhile, we modify the transformer encoder to train the FFR module such that it can successfully reconstruct the masked frame feature $h_t$ of any given video snippet. During inference, our system tries to use FFR to reconstruct a frame feature as $h_t'$, and the reconstruction error $h_t − h_t'$ is exploited to tell whether it is a boundary frame. This process is similar to humans, as humans are good at anticipating intra-event content rather than the transitions between two events. Likewise, due to its heterogeneous local

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
context, the FFR module is usually worse at reconstructing event boundary frames.

During the learning process, the objective function of our CoSeg is to minimize the following loss:

$$\mathcal{L} = \mathcal{L}_C + \beta \ast \mathcal{L}_R$$

where $\mathcal{L}_C$ is the loss for CTFE (see Section III-A), and $\mathcal{L}_R$ is the loss applied to the FFR (see Section III-B).

### A. Contrastive Temporal Feature Embedding

In general, a video event consists of a sequence of semantically correlated frames. Namely, the neighbor frames are more likely to be semantically similar than frames sampled at long time intervals. In line with this observation, we propose a CTFE scheme to learn a discriminative frame representation. Essentially, it projects the semantically similar frames closer, while pushing away the dissimilar ones. Leveraging the idea of contrastive learning, our framework transforms frames into a new representation being more semantically distinguishable.

As illustrated in Fig. 3, the positive pairs of contrastive learning consist of intra-segment frames, and the negative pairs come from inter-segment frames of other snippets from the same or other videos, or frames in the memory.

More specifically, we select a batch of $B$ videos from a non-curated video dataset, and then we randomly extract $X$ non-overlapped video snippets of $T$ frames from each video. It results in $L = B \ast X$ video snippets in total. Let $x_{ij}^k$, the $j$th frame of snippet $s_i$, be a query frame. We select the neighbor frames of query $x_{ij}^k$ in $s_i$ to form positive pairs. This is based on the assumption of the temporal semantic homogeneity of an event. Next, we form three types of negative pairs associated with query $x_{ij}^k$: 1) intravideo negative pairs: the negative frame comes from the same video as $x_{ij}^k$ but from a different snippet; 2) inter-video negative pairs: the negative frame is selected from any snippet $s_l$ extracted from a different video of $s_i$; and 3) memory negative pairs: the negative frame comes from frames embedded in the memory during previous iterations. Note that we only store one embedded frame per video segment in the memory, a first-in-first-out (FIFO) queue with the size of $K$. This design can greatly increase the number of representative frames to boost its performance. During training, our goal is to increase the similarity of the embedding of positive pairs but decrease that of negative pairs by minimizing the following loss function:

$$\mathcal{L}_C = \mathbb{E}_{i \in [0, L-1], j \in [0, T-1]} \left[ -\frac{1}{T-1} \sum_{k=0, k \neq j}^{T-1} \log P(i, j, k) \right]$$

with

$$P(i, j, k) = \frac{Q^+(i, j, k) + Q^-_1(i, j) + Q^-_2(i, j)}{\sum_{l=0, l \neq j}^{T-1} \sum_{m=0}^{K-1} \exp(\text{sim}(h_{lj}^m, z_n)/\tau)}$$

$$Q^+(i, j, k) = \exp(\text{sim}(h_{ij}^k, z_n)/\tau)$$

$$Q^-_1(i, j) = \sum_{l=0, l \neq j}^{T-1} \sum_{m=0}^{K-1} \exp(\text{sim}(h_{lj}^m, z_n)/\tau)$$

$$Q^-_2(i, j) = \sum_{n=0}^{K-1} \exp(\text{sim}(h_{ij}^k, z_n)/\tau)$$

where $\text{sim}$ denotes the cosine similarity; $Q^+(i, j, k), Q^-_1(i, j),$ and $Q^-_2(i, j)$ are the exponential temperature smoothed similarities of positive pairs, and the sum of exponential temperature smoothed similarities of inter-video and intravideo
negative pairs and negative pairs from memory, respectively; $h$ and $z$ are the corresponding features by the query encoder $f_q$ and the key encoder $g_k$, respectively; $\tau$ is the temperature set to 0.2, and $L$ is the total number of video snippets in the batch. Here, the query and key encoder architecture are widely used in contrastive learning; refer to [28] for details.

### B. Frame Feature Reconstruction

As we know, the transition frames between video events are usually inconsistent and thus less predictable. Accordingly, we develop an unsupervised feature reconstruction approach to detect these event boundaries, because we conjecture that boundary frames usually carry higher reconstruction errors than non-boundary ones. Unlike previous pixel-level image reconstruction [14], however, our frame reconstruction is conducted in a high-level semantic feature space. Namely, our approach aims at reconstructing a frame’s semantic representation trained by CTFE (Section III-A).

1) **Module Input:** Let us assume $x_t$ is the masked frame to be reconstructed from its local context of segment $s$ with $T$ frames. By applying feature extractor $f_{o}$ to its frames, we obtain the video segment’s semantic representation as $H = \{h_{0}, \ldots, h_{T-1}\}$. When training the reconstruction network, $h_t$ is a masked element, while during inference it should be reconstructed from its local context. In addition, we apply position embedding to all the frames of $s$ as $E_{pos} \in \mathbb{R}^{T \times D} = \{pos_0, \ldots, pos_i, \ldots, pos_{T-1}\}$, where $pos_i \in \mathbb{R}^D$ is the sin–cos position embedding for frame $x_t$, defined as follows:

$$
pos_i = \begin{cases} 
\sin(w_k \ast t), & i = 2k \\
\cos(w_k \ast t), & i = 2k + 1, \\
\end{cases} \quad w_k = \frac{1}{10000^{2k/D}}$$

where $D$ is the feature dimension, and $pos_i$ is the $i$th element of vector $pos_i$. Afterward, we define the module input as follows:

$$H_0(t) = \{h_0, \ldots, [\text{MASK}], \ldots, h_{T-1}\} + \{pos_0, \ldots, [\text{MASK}], \ldots, pos_{T-1}\}$$

(2) where $[\text{MASK}]$ denotes the corresponding token is masked at position $t$.

2) **Reconstruction Architecture:** To reconstruct the masked feature vector from $H_0(t)$, we modify the multilayer attention part of the transformer encoder. Specifically, we adopt two layers of multilayer self-attention (MSA) and multilayer perceptron (MLP) blocks [43] to process $H_0$ while randomly applying the mask $M(t)$ to the $i$th feature embedding. The output of the $l$th layer of the reconstruction module is defined as follows:

$$\begin{align*}
H'_l & = \text{MSA}(\text{LN}(H_{l-1}), M(t)) + H_{l-1} \\
H_l & = \text{MLP}(\text{LN}(H'_l)) + H'_l
\end{align*}$$

(3) where $H_{l-1}$ is the output of the last $l-1$th layer, and $\text{LN}$ is the layer normalization [46].

3) **Reconstruction Objective:** Previous work [44] formulates the mask reconstruction as a “masked language model” (MLM) problem, which attempts to optimize the entropy of similarity between the reconstructed token and the original one [47]. However, the embedded features of consecutive frames are very similar, which is not suitable for entropy loss. Therefore, we directly treat it as an absolute feature reconstruction problem instead. To enforce the model to learn a local temporal relationship, we use the mean-squared-error (MSE) loss to measure the reconstructed feature

$$L_R = \mathbb{E}_{t \sim V, r \sim [0, ..., T-1]} \| h'_t - h_t \|_2^2$$

(4) where $V$ is the overall video dataset, $h'_t$ is the reconstructed feature for the masked $t$th frame, and $h_t$ is its original one.

### C. Inference of CoSeg

As shown in the cognitive experiments [15], a new event is triggered once people fail to anticipate it. Likewise, frames within an event are easier to be reconstructed than boundary ones. This is reflected by the changes in the reconstruction errors. Hence, we apply the trained CoSeg system to detect event boundaries using the reconstruction errors as illustrated in Fig. 2(b).

Here, it is worth to mention that we used reconstruction loss to optimize the FFR module while we also used the reconstruction error for boundary detection. First, most frames in a video are consistent and relatively easier to reconstruct. Therefore, during training, most of our training data are non-boundary cases. That indicates our model tends to learn the general reconstruction pattern while boundary reconstruction is not learned too much. That is also similar to cognitive observations, and human online memory easily keeps track of intra-event observations while event boundary detected due to failure of new event reconstruction. Second, though our model is trained to reconstruct frames, the boundary frame is much harder to reconstruct compared with that of the common frames, which means its reconstruction error is likely to be higher in the boundary region. That phenomenon helps human to detect the boundaries and also help CoSeg to use the reconstruction error to detect event boundaries.

During event boundary detection, CoSeg scans a video sequentially from its beginning to the end with a window of $T$ frames corresponding to a $T$-length video snippet or local context. Given a video snippet, we apply the FFR to its middle frame, and then obtain a reconstruction error for each frame to form an error trajectory. Next, to detect boundaries from the error trajectory, we conduct the signal processing steps as follows (see Fig. 4).

1) **Filter Processing:** The FIR filter $\tilde{S}[t] = \sum_{k=N}^{t} e_k E[t-k]$, where $e_k$ is the coefficient, is adopted to process the time-series signal $E$, which can reduce the noise and enhance the prominent features of the time series at the same time. Here, we set $e_k = (1/2N + 1)$ to smooth the error trajectory.

2) **Gradient Calculation:** Calculate gradient $G$ from $S$. 

3) Relative Extrema Detection: The relative extrema detection [48] is applied by picking up timestamp \( t_b \) as a boundary by the following equation:

\[
\begin{align*}
G[t_b] > G[t], & \quad t \in [t_b - r, t_b - 1] \\
G[t_b] > G[t], & \quad t \in [t_b + 1, t_b + r]
\end{align*}
\] (5)

where \( r \) is the range for relative extrema detection.

IV. EXPERIMENTS

A. Implementation Details

We adopt ResNet-18 [50] as the backbone of the feature extractor and modify the transformer encoder [43] with two layers of eight attention heads as the reconstruction architecture. Here, we use ResNet-18 rather than other powerful architectures to have a fair comparison with previous event segmentation approaches. In contrastive feature embedding, the segment size \( T \) is 10, the temperature \( \tau \) is set as 0.2, and the queue size \( K \) of memory is 65,536. The key encoder \( g_0 \) is the momentum update of the query encoder with \( g_0 = \alpha \cdot g_0 + (1-\alpha) \cdot f_s \), where \( \alpha \) is set to 0.999. In temporal feature reconstruction, we mask one feature embedding to reconstruct it. The coefficient \( \beta \) of overall loss in (1) is simply set to 1. We use the SGD [51] optimizer to optimize the overall framework with the learning rate 0.002, weight decay 1e-4, and momentum 0.9. The training batch size is 16, and the number of video snippets from one video \( c \) is 2, resulting in 32 video snippets in total. All the experiments are done in a server with 4 NVIDIA Tesla P40 GPUs.

B. Datasets

Following the same protocol of previous work [11], [12], [13], [14], we evaluated CoSeg on the following public datasets: Breakfast [3], INRIA instructional [7], 50 Salads [21], and GEBD [20].

1) Breakfast Dataset: consists of 1989 videos of ten breakfast activities, where one person may perform different activities in different videos. The video qualities vary a lot due to occlusions and different viewpoints.

2) INRIA Instruction Dataset: mainly contains 2-min-long videos collected from YouTube. There are about 47 sub-activities in this dataset. This dataset is very challenging for self-supervised event segmentation approaches because it carries a substantial number of “background” scenes, which generally do not share obvious common visual patterns.

3) 50 Salads Dataset: collects multimodal data from the cooking scenarios. In this dataset, the event segmentation is specified at different levels of granularity—high, low, and eval. Following previous self-supervised works [5], [22], [27], we use the “eval” granularity to train and evaluate our framework.

4) Kinetics-GEBD: is the newest unconstrained video dataset specifically developed for the evaluation task of GEBD. It carries the largest number of event boundaries (e.g., 32 \times ActivityNet and 8 \times EPIC-Kitchens-100) from videos “in the wild,” covering boundaries caused by action changes or generic event changes. Unlike other event segmentation datasets, such as Breakfast, INRIA Instruction, and 50 Salads, collected from a predefined taxonomy, Kinetics-GEBD is an open-vocabulary dataset. It consists of the train/val/test set with each having about 20 K videos randomly sampled from the corresponding set of Kinetics-400 dataset [20], respectively. The Kinetics-GEBD dataset labels the event boundaries from the following five different high-level causes: 1) change in subject; 2) change in object of interaction; 3) change in action; 4) change in environment; and 5) shot change.

C. Evaluation Metrics

As aforementioned, our goal is to detect generic event boundaries from videos “in the wild,” where a boundary is defined as a timestamp or a short-range represented by its middle timestamp. To evaluate the detection algorithm, the discrepancy between the detected boundary and the ground-truth boundary is the most important measurement for us. In this work, we adopt the Precision/Recall and F1 score as our primary evaluation metrics. Specifically, our evaluation formula is similar to that of the GEBD challenge held at CVPR 2021 [18]. Just like the intersection-over-union (IoU) measurement, the relative distance (Rel.Dis.) is calculated as the error between the detected and ground-truth boundary, divided by the length of the corresponding video instance. Given a fixed threshold of Rel.Dis. (e.g., 5% in our evaluation), the Precision/Recall and F1 score are computed by determining whether a detection is correct or not within the threshold. Note that the F1 score is also widely used in supervised event segmentation tasks [5], [22]. These evaluation metrics can be evaluated in a matching-free style, which can provide a more direct and fair comparison across different methods.

In addition, we also try some additional evaluation metrics including the mean over frames (MoF) and IoU, which are widely used to evaluate the (weakly) supervised or clustering-based unsupervised nongeneric event segmentation works [11], [12], [13], [14]. To calculate these metrics, the Hungarian matching algorithm needs to be applied to build a one-to-one match between the predicted segments and the ground-truth event segments. This matching is conducted for a set of predefined event categories. However, since our work aims at the generic event boundary regardless of its event category, we are unable to perform global matching like [13]. Instead, we apply matching at the video level. In other words, as we do not assign an event label or pseudolabel (clustering-based approaches) to the segments, our matching varies by video. Therefore, we should be aware that we cannot directly compare our approach with previous works by means of MoF and IoU. It is just for reference to list the results in terms of these metrics.

Let us assume there are \( U \) segments in the ground truth, and consider a matched pair \((Y, Z)\) where \( Y \) is a predicted segment and \( Z \) is the ground truth. Furthermore, let \( Y \cap Z \) denote the intersection of two segments and \( Y \cup Z \) represent their union. Then the metrics MoF and IoU can be defined as follows:

\[
\begin{align*}
\text{MoF} & = \frac{\sum_{k=1}^{U} Y_k \cap Z_k}{\sum_{k=1}^{U} Z_k} \\
\text{IoU} & = \frac{1}{U} \sum_{k=1}^{U} \frac{Y_k \cap Z_k}{Y_k \cup Z_k}
\end{align*}
\] (6)
D. Performance Analysis on CoSeg

1) Performance at Different Rel.Dis. Thresholds: The primary evaluation metric for generic event detection is F1 scores (calculated from Precision/Recall) at a predefined Rel.Dis. threshold by users. In general, a smaller threshold implies a higher criterion to determine whether a boundary detection is correct. In other words, the error between the detected boundary and the ground truth is smaller. Table I displays the event boundary detection F1 scores on the Kinetics-GEBD dataset given various Rel.Dis. thresholds. When the criterion is getting easier (i.e., higher Rel.Dis. threshold), the F1 score is increasing. We also note that when the threshold is over 0.2, the change in the F1 score is minor. In the rest of our experiments, we only report the Precision/Recall/F1 score at 5%, which is the strictest criterion to calculate F1. As we can observe, we are even higher than those supervised methods. That is because they are very standard supervised methods without introducing complicated mechanisms and designs to optimize the network. Meanwhile, one big limitation here is Kinetics-GEBD is a more challenging dataset which limits direct application of traditional methods for event segmentation. With reasonable adjustment, supervised or weakly supervised methods may achieve better performance.

2) Effectiveness of the CTFE Module: In this group of experiments, we want to verify the effectiveness of the proposed CTFE (see Section III-A) in our CoSeg system. To this end, we replaced the self-trained CTFE module with an ImageNet pretrained visual geometry group (VGG) [52] (ImageNet-VGG) or ResNet [50] (ImageNet-RN) encoder. All the relevant experiments were conducted on the Breakfast dataset with default settings. Table II illustrates their performance comparisons in terms of various evaluation metrics. It clearly shows that the CTFE approach can outperform both the ImageNet pretrained models by a large margin in all the evaluation metrics, such as by 2%–3% in MoF, 1%–2% in IoU, and 16%–17% in F1@5%. These improvements actually confirm our conjecture that our CTFE can generate better feature representation for event segmentation, thanks to its capability of capturing the underlying temporal difference between frames of the inter-event and intra-event. In addition, note that unlike the ImageNet pretrained models, our feature embedding does not require to be trained with external datasets, which makes the end-to-end CoSeg framework more flexible.

3) Effectiveness of the FFR Module: We further conducted a set of experiments to verify the effects of various settings for our FFR module (see Section III-B). The experimental results on the Breakfast dataset are shown in Table III. The three columns (MX) in the left panel of the table display the results in terms of various metrics achieved by different reconstruction sizes (i.e., mask size, the number of frames to be reconstructed, 1/3/5 in our experiments) when fixing the window size $T = 10$. As we can see, the mask size of 1 (M1 column) achieves the best performance. It seems simultaneously reconstructing more than one frame may decrease the event boundary detection accuracy. Meanwhile, we also examined how the value of video snippet length $T$ (i.e., the size of the local context or window) affects boundary detection. In terms of all the evaluation metrics, $T = 10$ consistently outperforms other snippet lengths/window sizes. It is worth noting that adding a longer window size (local supportive context) does not necessarily improve the detection accuracy. This observation is consistent with the fact that humans can quickly detect a new event without knowing many contexts.

4) Generalizability of CoSeg Framework: Although long-term memory is built over various past experiences, humans are able to detect event boundaries under a completely

| Supervision | Methods          | 0.05 | 0.1  | 0.15 | 0.2  | 0.25 | 0.3  | 0.35 | 0.4  | 0.45 | 0.5  | avg  |
|-------------|-----------------|------|------|------|------|------|------|------|------|------|------|------|
| Supervised  | BMN [49]        | 0.186| 0.204| 0.213| 0.220| 0.226| 0.230| 0.233| 0.237| 0.239| 0.241| 0.223|
|             | BMN-StarEnd [49]| 0.491| 0.589| 0.627| 0.648| 0.660| 0.668| 0.674| 0.678| 0.681| 0.683| 0.640|
|             | TCN-TAPOs [5]   | 0.464| 0.560| 0.602| 0.628| 0.645| 0.659| 0.669| 0.676| 0.682| 0.687| 0.627|
|             | TCN [5]         | 0.588| 0.657| 0.679| 0.691| 0.698| 0.703| 0.706| 0.708| 0.710| 0.712| 0.685|
|             | PC [18]         | 0.625| 0.758| 0.804| 0.829| 0.844| 0.853| 0.859| 0.864| 0.867| 0.870| 0.817|
| Unsupervised| SceneDetect †   | 0.275| 0.300| 0.312| 0.319| 0.324| 0.327| 0.330| 0.332| 0.334| 0.335| 0.318|
|             | PA-Random [18]  | 0.336| 0.435| 0.484| 0.512| 0.529| 0.541| 0.548| 0.554| 0.558| 0.561| 0.506|
|             | PA [18]         | 0.396| 0.488| 0.520| 0.534| 0.544| 0.550| 0.555| 0.558| 0.561| 0.564| 0.527|
|             | CoSeg           | 0.656| 0.758| 0.783| 0.794| 0.799| 0.803| 0.804| 0.806| 0.807| 0.809| 0.782|

| Methods | MoF | IoU | Prec@5% | Recall@5% | F1@5% |
|---------|-----|-----|---------|-----------|-------|
| ImageNet-VGG | 49.2 | 47.4 | 41.4 | 35.4 | 38.2 |
| Replacement |        |     |       |        |       |
| ImageNet-ResNet | 50.7 | 47.3 | 43.0 | 35.5 | 38.9 |
| Replacement |        |     |       |        |       |
| Our CTFE CoSeg | 53.1 | 48.6 | 47.0 | 65.4 | 54.7 |

| mask size $M$ | Input len $T$ | M1 | M3 | M5 | T5 | T10 | T20 | T30 |
|---------------|---------------|----|----|----|----|-----|-----|-----|
| MoF           |               | 53.1| 53.0| 52.8| 51.6| 53.1| 50.9| 51.1 |
| IoU           |               | 48.6| 47.3| 45.7| 46.7| 48.6| 47.5| 47.8 |
| Prec@5%       |               | 47.0| 48.4| 47.6| 45.8| 47.0| 47.2| 47.3 |
| Recall@5%     |               | 65.4| 61.9| 63.4| 63.3| 65.4| 63.2| 63.4 |
| F1@5%         |               | 54.7| 54.3| 54.3| 53.1| 54.7| 54.0| 54.1 |
| Infer Time (s) |               | 53.3| 51.4| 48.9| 47.3| 53.3| 58.8| 62.0 |
TABLE IV
Transfer Learning Performance of CoSeg

| Target    | Source    | MoF | IoU | Prec@5% | Recall@5% | F1@5% |
|-----------|-----------|-----|-----|---------|-----------|-------|
| Breakfast | Breakfast | 53.1| 48.6| 47.0    | 65.4      | 54.7  |
| 50salad   | INRIA     | 51.9| 48.1| 46.0    | 65.8      | 54.2  |
| 50salad   | INRIA     | 47.9| 42.4| 46.7    | 63.3      | 53.7  |
| Breakfast | INRIA     | 45.7| 42.2| 45.2    | 66.6      | 53.9  |
| 50salad   | INRIA     | 45.9| 43.0| 45.6    | 66.2      | 54.0  |
| Breakfast | 50salad   | 64.1| 42.3| 63.7    | 82.1      | 71.8  |
| 50salad   | INRIA     | 60.7| 42.0| 56.9    | 90.0      | 69.7  |
| 50salad   | INRIA     | 62.9| 41.9| 57.1    | 90.7      | 70.0  |

new situation. This is the capability of generalization. Likewise, can we train the CoSeg with different datasets and test it on an unseen dataset? To answer this, we designed a set of cross-dataset experiments, in which we trained the CoSeg model from the source dataset and apply the model for event segmentation on the target dataset. All the results are shown in Table IV. We note that our CoSeg has a strong capability of generalization across datasets. In other words, the CoSeg model trained on different datasets can achieve similar performance. In particular, the INRIA Instructional dataset looks very different from the Breakfast and 50 Salads datasets, but the performance of cross-dataset transfer is still promising. These experiments tell us that the generalizability of CoSeg enables our model to detect more generic event boundaries from videos “in the wild.”

5) Sensitivity to Error Detection Hyperparameter: To investigate the influence of hyperparameters of error detection, we examine the performance on the Breakfast dataset with different “range” parameters $r$ [see (5)] to detect the extrema as an event boundary. As shown in Table V, our performance is stable under different hyperparameters for detecting the event boundaries. Here, smaller $r$ leads to smaller MoF. That indicates a smaller $r$ has more segments and shows less preference for the dominant classes, while higher $r$ demonstrates preferences for dominant classes and results in fewer segments. But different $r$ always yields higher values on different metrics. To conclude, our extraordinary performance is not dependent on the error detection hyperparameters.

E. Performance Comparisons to SOTA Approaches

The task of video event segmentation has been developed for many years, which results in various benchmark datasets and evaluation metrics. In this section, we will compare the performance of our CoSeg with other state-of-the-art (SOTA) approaches on different benchmark datasets with five evaluation metrics.

1) Kinetics-GEBD Dataset: As aforementioned, Kinetics-GEBD is the most challenging and largest benchmark for GEBD. We first want to demonstrate how well our CoSeg performs on this dataset when compared with other approaches. We trained the models on the train set and tested them on val set (the test set is not publically available). Table VI presents the F1 scores of event boundary detection at different Rel.Dis thresholds from our CoSeg system and other SOTA approaches. As can be seen, our CoSeg outperforms all the unsupervised approaches by a large margin. In terms of average F1 scores at all the thresholds, our F1 score is about 26% better than the second best. Our approach is even better than most of the supervised approaches as reported in [18] and is comparable to the best supervised approach Pairwise boundary Classifier (PC) [18] (only about 4% lower in average F1 scores than the PC method).

2) Breakfast Actions Dataset: This dataset was prepared for the evaluation of previous supervised and cluster-based unsupervised event segmentation, which deals with a set of predefined event categories rather than generic events. The MoF and IoU metrics are widely used to evaluate the predefined event segmentation algorithms. To compare with previous results, we also calculated both the metrics on this dataset. However, as discussed in IV-C, we are unable to directly compare our method with previous works in MoF and IoU. To make a fair comparison, we also calculated the Precision/Recall/F1 score @5% threshold for this dataset.

Table VII displays all the results in all the five metrics. Most previous works do not report the Precision/Recall/F1 scores, and we are unable to acquire their source code. Hence, we will leave these metric values empty for this work. It is obvious that CoSeg outperforms all the previous work in all the evaluation metrics. The comparison between our work and LSTM+AL is meaningful in terms of MoF and IoU, as both...
TABLE VII
SEGMENTATION RESULTS ON THE BREAKFAST DATASET

| Supervision | Methods       | MoF | IoU | Prec@5% | Recall@5% | F1@5% |
|-------------|---------------|-----|-----|---------|-----------|-------|
| Unsupervised| KNN+GMM [13]  | 34.6| 47.1| -       | -         | -     |
|             | CTE-MLP [12]  | 41.8| -   | 32.5    | 30.3      | 31.3  |
|             | U-Net [14]    | 42.7| 12.8| -       | -         | 29.9  |
|             | LSTM+AL [27]  | 42.9*| 46.9*| -       | -         | -     |
|             | CoSeg         | 53.1*| 48.6*| 47.0    | 65.4      | 54.7  |

* indicates the metric calculation without global matching

TABLE VIII
SEGMENTATION RESULTS ON THE INRIA DATASET

| Supervision | Methods       | MoF | IoU | Prec@5% | Recall@5% | F1@5% |
|-------------|---------------|-----|-----|---------|-----------|-------|
| Unsupervised| KNN+GMM [13]  | 27.0| -   | -       | -         | -     |
|             | CTE-MLP [12]  | 39.0| 9.6 | 61.6    | 18.5      | 28.4  |
|             | U-Net [14]    | 39.1| 9.4 | -       | -         | 29.9  |
|             | CoSeg         | 47.9*| 42.4*| 46.7    | 63.3      | 53.7  |

TABLE IX
SEGMENTATION RESULTS ON THE 50 SALADS DATASET

| Supervision | Methods       | MoF | IoU | Prec@5% | Recall@5% | F1@5% |
|-------------|---------------|-----|-----|---------|-----------|-------|
| Unsupervised| CTE-MLP [12]  | 35.5| -   | 74.9    | 36.5      | 49    |
|             | LSTM + KNN [11]| 54.0| -   | -       | -         | -     |
|             | LSTM+AL [27]  | 60.6*| -   | -       | -         | -     |
|             | U-Net [14]    | -   | -   | -       | -         | 56.9  |
|             | CoSeg         | 64.1*| 42.3*| 63.7    | 82.1      | 71.8  |

do not have global matching (marked with * in the table). A Higher IoU value indicates CoSeg does not suffer from the oversegmentation problem even with a higher MoF value. In terms of the F1@5% score, our work is significantly better (over 20% higher) than the latest U-Net and CTE-MLP work, which is very promising. It further proves the effectiveness of our end-to-end self-supervised framework.

3) INRIA Instructional Video Dataset: INRIA dataset is more challenging than the Breakfast dataset due to its widespread “Background” segments. We repeated the evaluation of Breakfast onto the INRIA dataset. The comparisons to other approaches are listed in Table VIII. Most previous approaches are biased due to their low IoU scores, which suggests those methods prefer dominant classes instead of finding the correct segmentation. In contrast, our CoSeg achieves the best performance in all the metrics, and it can keep a good balance between dominant and underrepresented events. The superior results have shown our self-supervised framework integrated with long-term and online memory can deal with the “Background” segments and detect proper segmentation with much less undersegmentation. Most importantly, we yield 63.3% recall of ground-truth boundary while CTE-MLP only recalled 18.5%, which means a large percentage of boundaries are missing.

4) 50 Salads Dataset: 50 Salads is a multimodality dataset, but we only used its RGB modality in our experiments. Following the experiments on the previous two datasets, we illustrate the experimental results on 50 Salads in Table IX. As can be seen, CoSeg can still significantly outperform the recent U-Net and CTE-MLP in F1 score @5%. Under the same MoF calculation, our approach can still beat the LSTM+AL approach.

In all the four datasets, our framework greatly improves the performance without using annotated data or external data. Meanwhile, compared with previous unsupervised methods, we achieved higher MoF and IoU at the same time. It suggests that our method alleviates the overfitting problem which widely existed in previous methods. In addition, a much higher F1@5% also suggests that the segmentation results of CoSeg have fewer over- or undersegmentation instances.

F. Qualitative Evaluation

In this section, we will demonstrate and analyze some qualitative results by visualizing the event segmentation results of our CoSeg on the Breakfast and 50 Salads datasets.

Fig. 5 illustrates the visualization comparisons of event segmentation results for a video (P33_cam03_cereals) from the Breakfast dataset using a different model. Fig. 5(a) displays three t-stochastic neighbor embedding (t-SNE) visualization results of visual features learned by VGG, ResNet, and our CoSeg network, respectively, where each color represents one event segment/cluster. As can be seen from t-SNE visualization, CoSeg demonstrates much better feature aggregation than other pretrained networks, e.g., the green and blue features corresponding to the event “pour cereals” (segment 1) and “pour milk” (segment 2). It visually verifies that similar visual features can be easier to be clustered under the self-supervised mechanism. Fig. 5(b) further illustrates the event segmentation results of our CoSeg and its variants (the replacement with ImageNet-VGG or ImageNet-ResNet). We can observe that the segmentation of CoSeg clearly shows better matching to the...
problems for complicated scenarios that occur in long videos. Alternative embedding solutions. Fewer segments detected by embedding module can achieve better aggregation than other approaches (nonmatch segments are shown in light-green).

Fig. 5. Qualitative analysis of CoSeg on the Breakfast dataset [3]. This example is from P33_cam03_cereals video. (a) Compares the t-SNE [53] of the extracted visual features by the ImageNet-pretrained VGG and ResNet network, as well as our CoSeg. Self-supervised visual features show a better clustering in terms of the “Pour cereals” and “Pour milk” events. It can even group the frames without meaningful actions with the knowledge learned from the entire dataset (shown in red color). (b) Our segments agree well with the ground-truth labels, even though the overall background and visual conditions remain the same, which increases the challenge for detecting events. Other models fail to segment segments 0 and 1 due to their weak visual representations.

ground truth. Although the features extracted from segments 2 and 3 are intertwined with each other, our CoSeg can still detect the event boundary between segments 2 and 3, thanks to the advancement of the FFR module.

Fig. 6 shows another visualization example from the 50 Salads dataset. This example is from rgb-22-1 video of the 50 Salads dataset and is a good example to show an oversegmentation problem for long videos. (a) t-SNE [53] of the extracted visual features by the ImageNet-pretrained VGG and ResNet network, as well as TCSE of CoSeg. (b) Temporal segmentation timelines of different approaches (nonmatch segments are shown in light-green).

V. CONCLUSION

In this article, we propose a self-supervised framework, called CoSeg, for GEBD or event segmentation. The system design is inspired by some recent cognitive research on how humans detect new events. First, we invent a CTFE strategy to acquire semantic meaningful feature representation, which mimics the function of a human’s long-term memory. Second, we formulate the event segmentation problem as an FFR task, which is similar to the behavior of humans predicting new events. We thoroughly evaluate our framework on four datasets. The experiments demonstrate that our framework can outperform other solutions by a big margin. Furthermore, compared with previous works, no prior information is needed for CoSeg so that it can be readily applied to new datasets with competitive performance. Most notably, CoSeg has achieved SOTA results on the GEBD dataset, which clearly prove its generalizability on general event boundary detection tasks.

ACKNOWLEDGMENT

The authors would like to thank Rosaura’s kind help in evaluating their method Unet [14] with our new evaluation metrics. They also would like to thank Yuanyuan Zhang for her help with figure revising.

REFERENCES

[1] J. M. Zacks and B. Tversky, “Event structure in perception and conception,” Psychol. Bull., vol. 127, no. 1, pp. 3–21, 2001.
[2] C. Xu and L. Ding, “Weakly-supervised action segmentation with iterative soft boundary assignment,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6508–6516.
[3] H. Kuehne, A. Arslan, and T. Serre, “The language of actions: Recovering the syntax and semantics of goal-directed human activities,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 780–787.
[4] H. Kuehne, J. Gall, and T. Serre, “An end-to-end generative framework for video segmentation and recognition,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–8.
[5] C. Lea, A. Reiter, R. Vidal, and G. D. Hager, “Segmental spatiotemporal CNNs for fine-grained action segmentation,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 36–52.
[6] A. Richard, H. Kuehne, and J. Gall, “Weakly supervised action learning with RNN based fine-to-coarse modeling,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 754–763.
[7] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien, “Unsupervised learning from narrated instruction videos,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4575–4583.
[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.
[9] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy, “What’s Cookin’? Interpreting cooking videos using text, speech and vision,” 2015, arXiv:1503.0538.
[10] P. Bojanowski et al., “Weakly supervised action labeling in videos under ordering constraints,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 628–643.
[11] B. L. Bhatnagar, S. Singh, C. Arora, C. Jawahar, and K. CVIT, “Unsupervised learning of deep feature representation for clustering egocentric actions,” in Proc. IJCAI, 2017, pp. 1447–1453.
[12] A. Kukleva, H. Kuehne, F. Sener, and J. Gall, “Unsupervised learning of action classes with continuous temporal embedding,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12066–12074.
[13] F. Sener and A. Yao, “Unsupervised learning and segmentation of complex activities from video,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8368–8376.
[14] R. G. VidalMata, W. J. Scheirer, A. Kukleva, D. Cox, and H. Kuehne, “Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences,” 2020, arXiv:2001.11122.
WANG et al.: CoSeg: COGNITIVELY INSPIRED UNSUPERVISED GENERIC EVENT SEGMENTATION 12517

[15] C. A. Kurby and J. M. Zacks, “Segmentation in the perception and memory of events,” Trends Cogn. Sci., vol. 12, no. 2, pp. 72–79, Feb. 2008.

[16] J. M. Zacks and K. M. Swallow, “Event segmentation,” Current Direc tions Psychol. Sci., vol. 16, no. 2, pp. 80–84, 2007.

[17] J. M. Zacks, N. K. Speer, J. M. Vettel, and L. L. Jacoby, “Event understanding and memory in healthy aging and dementia of the Alzheimer type,” Psychol. Aging, vol. 21, no. 3, pp. 466–482, Sep. 2006.

[18] M. Z. Shou, S. W. Lei, W. Wang, D. Ghadiyaram, and M. Feiszli, “Generic event boundary detection: A benchmark for event segmentation,” 2021, arXiv:2101.10511.

[19] D. Shao, Y. Zhao, B. Dai, and D. Lin, “Intra- and inter-action understanding via temporal action parsing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 730–739.

[20] W. Kay et al., “The kinetics human action video dataset,” 2017.

[21] M. Z. Shou, S. W. Lei, W. Wang, D. Ghadiyaram, and M. Feiszli, “Generic event boundary detection: A benchmark for event segmentation,” 2021, arXiv:2101.10511.

[22] T. Han, W. Xie, and A. Zisserman, “Memory-augmented dense predictive coding for video representation learning,” 2020, arXiv:2008.01065.

[23] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, “Connectionist temporal classification for video representation learning,” 2020, Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9729–9738.

[24] T. Han, W. Xie, and A. Zisserman, “Self-supervised co-training for video representation learning,” 2020, Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1197–1206.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9729–9738.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020, arXiv:2002.05709.

[27] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrastive cluster assignments,” 2020, arXiv:2006.09882.

[28] X. Wang, D. Kihara, J. Luo, and G.-J. Qi, “EnAET: A self-trained framework for semi-supervised and supervised learning with ensemble transformations,” 2019, arXiv:1911.09265.

[29] J. Donahue, P. Kr¨ahem¨abili, and T. Darrell, “Adversarial feature learning,” 2016, arXiv:1605.09782.

[30] J. Donahue and K. Simonyan, “Large scale adversarial representation learning,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 10542–10552.

[31] Y. M. Asano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” 2019, arXiv:1911.05371.

[32] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 132–149.

[33] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2006, pp. 1735–1742.

[34] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3733–3742.

[35] J.-B. Grill et al., “Bootstrap your own latent: A new approach to self-supervised learning,” 2020, arXiv:2006.07733.

[36] Q. Hu, X. Wang, W. Hu, and G.-J. Qi, “AdCo: Adversarial contrast for efficient learning of unsupervised representations from self-neglected adversaries,” 2020, arXiv:2011.08435.

[37] T. Han, W. Xie, and A. Zisserman, “Self-supervised co-training for video representation learning,” 2020, arXiv:2010.09709.

[38] M. Fayyaz and J. Gall, “SCT: Set constrained temporal transformer for set supervised action segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 501–510.

[39] T. Han, W. Xie, and A. Zisserman, “Memory-augmented dense predictive coding for video representation learning,” 2020, arXiv:2008.01065.

[40] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[42] T. B. Brown et al., “Language models are few-shot learners,” 2020, arXiv:2005.14165.

[43] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[44] A. Wang and K. Cho, “BERT has a mouth, and it must speak: BERT as a Markov random field language model,” 2019, arXiv:1902.04094.

[45] P. K. Vemulapalli, V. Monga, and S. N. Brennan, “Robust extreme features for time-series data analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 6, pp. 1464–1479, Jun. 2013.

[46] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, “BMN: Boundary-matching network for temporal action proposal generation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3889–3898.

[47] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 77–78.

[48] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in Proc. COMPSTAT, Cham, Switzerland: Springer, 2010, pp. 177–186.

[49] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[50] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, pp. 2579–2605, Nov. 2008.

Xiao Wang (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Computer Science, Purdue University, West Lafayette, IN, USA. His research interests include deep learning, computer vision, bioinformatics and intelligent systems. Dr. Wang is an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT VEHICLES.

Jingen Liu (Senior Member, IEEE) received the Ph.D. degree from the University of Central Florida, Orlando, FL, USA, in 2009. He is a Computer Vision Researcher at JD AI Research (JD.com Silicon Valley Research Center), Mountain View, CA, USA. Before joining JD.com, he was a Computer Scientist at SRI International, Menlo Park, CA, USA, from 2011 to 2018, and a Research Fellow at the University of Michigan, Ann Arbor, MI, USA, from 2010 to 2011.

Tao Mei (Fellow, IEEE) is the Technical Vice President with JD.com, Beijing, China. Dr. Mei was elected an International Fellow of the Canadian Academy of Engineering, a Fellow of IAPR and CAAI, and a Distinguished Scientist of ACM. He is a Senior/Associate Editor of IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, ACM Transactions on Multimedia Computing, Communications, and Applications.

Jiebo Luo (Fellow, IEEE) is a Professor with the Department of Computer Science, University of Rochester, Rochester, NY, USA. His research interests include computer vision, natural language processing (NLP), machine learning, data mining, multimedia computational social science and digital health.

Prof. Luo is a fellow of ACM, AAAI, SPIE, and IAPR. He has been Co-Chair of many conferences, as well as Editor-in-Chief or Editorial Board member for many journals.