Self-Regulated Learning for Egocentric Video Activity Anticipation

Zhaobo Qi, *Student Member IEEE,* Shuhui Wang, *Member IEEE,* Chi Su,
Li Su, Qingming Huang, *Fellow IEEE,* and Qi Tian, *Fellow IEEE*

Abstract—Future activity anticipation is a challenging problem in egocentric vision. As a standard future activity anticipation paradigm, recursive sequence prediction suffers from the accumulation of errors. To address this problem, we propose a simple and effective Self-Regulated Learning framework, which aims to regulate the intermediate representation consecutively to produce representation that (a) emphasizes the novel information in the frame of the current time-stamp in contrast to previously observed content, and (b) reflects its correlation with previously observed frames. The former is achieved by minimizing a contrastive loss, and the latter can be achieved by a dynamic reweighing mechanism to attend informative frames in the observed content with a similarity comparison between feature of the current frame and observed frames. The learned final video representation can be further enhanced by multi-task learning which performs joint feature learning on the target activity labels and the automatically detected action and object class tokens. SRL sharply outperforms existing state-of-the-art in most cases on two egocentric video datasets and two third-person video datasets. Its effectiveness is also verified by the experimental fact that the action and object concepts that support the activity semantics can be accurately identified.

Index Terms—Egocentric video activity anticipaiton, Third-person video activity anticipaiton, Contrastive learning, Multi-task learning, Self-regulated learning.

◆

1 INTRODUCTION

Egocentric perception has received remarkable research attention in recent years. An increasing number of tasks (*e.g.,* egocentric video summarization, egocentric localization, egocentric object detection, egocentric action recognition and anticipation) and benchmark datasets (*e.g.,* Ego-Sum+gaze [1], EGTEA Gaze+ [2], Charades-Ego [3] and EPIC-Kitchens [4]) are proposed. The construction of large-scale egocentric video datasets, such as EPIC-Kitchens, further promotes the technical advance in this field. Among a diversified range of egocentric vision tasks, anticipating future activities, which aims to predict what will possibly happen in the future, has become an active research topic due to its wide application prospectives. For example, in human-robot interaction scenario, robots can work closely with humans if they are able to anticipate human actions in the next few minutes [5], and in autonomous driving, an autonomous vehicle needs to anticipate if a pedestrian crosses the street and produce consequent system control

- *Corresponding author: Shuhui Wang and Qingming Huang.*
- *Z. Qi, L. Su and Q. Huang are with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China, and with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. Q. Huang is also with Peng Cheng Laboratory, Shenzhen 518066, China.*
  E-mail: zhaobo.qi@vipl.ict.ac.cn, {suli, qmhuang}@ucas.ac.cn.
- *S. Wang is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China.*
  E-mail: wangshuhui@ict.ac.cn.
- *C. Su is with Kingsoft Cloud, Beijing, 100085.*
  Email: suchi@kingsoft.com
- *Q. Tian is with Cloud BU, Huawei Technologies, Shenzhen 518129, China.*
  E-mail: tian.qi1@huawei.com.

(a) **Illustration of the problem.** Video delivers diversified content and rich context. Recursive sequence prediction model produces inaccurate intermediate representations.

(b) **The workflow of SRL.** The predicted intermediate representation is rectified by Revision, the importance of previously observed key frame feature is reweighed by Reattend, and the semantic context information (actions, objects) related to the target activity is used to enhance the target activity representation.

Fig. 1. **Illustration of the problem and the major workflow of SRL**.

command to guarantee driving safety [6].

Due to recent progress in egocentric vision, existing models can predict what will happen within a time horizon of up to several minutes [7], [8], [9]. Nevertheless, future activity anticipation is still a challenging task. An example is shown in Figure 1(a), as defined in [4], the 'anticipation time' τ_a is the time interval from the activity we need to anticipate, and the 'observation time' τ_o is the observed length of the video in advance to the target activity. The goal of the future activity anticipation task is to predict the

activity label of the video clip $[\tau_s, \tau_e]$ by observing a video clip $[\tau_s - (\tau_o + \tau_a), \tau_s - \tau_a]$, which precedes the target activity start time τ_s by τ_a.

A standard future activity anticipation paradigm is the recursive sequence prediction [7], [8], [10], where the anticipation model sees all observed video clips and predicts what will happen at the next clip. The process is repeated until the desired prediction moment τ_s is reached, as shown in Figure 1(a). Based on this process, the key of obtaining accurate activity prediction at the end depends on how to extract and represent the informative visual cues in the video content during the anticipation stage. In general, there are several crucial issues that need to be investigated.

First, activity videos contain drastic changes on both appearance and semantics from the beginning to the end. In Figure 1(a), the 'make breakfast' video contains a series of activities, *i.e.*, 'put pancake', 'take knife', 'open butter', 'grab butter', 'spread butter' and 'close butter'. These activities are represented by key frames with very different appearance, *i.e.*, the actions and objects inside a frame may be different from one to another. On the other hand, different activities demonstrate consecutive but diversified contextual dependencies. In Figure 1(a), from object perspective, 'butter' appears in frames of 'open butter', 'spread butter' and 'close butter'; from event perspective, 'spread butter' seems to have stronger contextual relation to 'take knife'.

At the beginning of recursive sequence prediction, we obtain an initial feature representation based on the observed video clip before $\tau_s - \tau_a$. In model training, only the initial feature representations and the current frame in anticipating stage can be used, so the obtained consecutive intermediate representations may be far less accurate. If the intermediate representations are directly used in subsequent prediction, the accumulation of errors may result in inaccurate final target activity prediction. To address the above issues, we propose a simple and effective Self-Regulated Learning (SRL) framework for future activity anticipation, which makes full use of the rich information only contained in the video to learn the anticipated representations in an unsupervised manner.

SRL aims to regulate the intermediate representation consecutively to produce representation that (a) emphasizes the novel information at the current time-stamp in contrast to previously observed content; and (b) reflects its correlation with previously observed frames. See Figure 1(b), the former requires a revision operation on the intermediate representation generated by the pre-trained sequential prediction model. At each anticipation time-step, a contrastive loss is utilized to rectify the predicted intermediate feature representation by treating the intermediate representation as positive sample and a batch of semantically uncorrelated frames as negatives sampled without difficulty. The latter demands a dynamic reweighing mechanism to attend to informative frames in the observed content, which resorts to a similarity comparison between feature of the current frame and observed frames. The highly similar frames may dominate the reweighed observed features, which, in combination with the revised representation, is fed into another sequential prediction unit to produce a sequence of more complete and accurate feature representations.

Compared to recent iterative prediction methods [7], [8],

[10], SRL can learn a representation that is less error-prone and avoids performance degradation due to accumulation of anticipation error. Compared to other works [9], [11] that directly utilize observed representations to anticipate long-term future activity without intermediate anticipation, the intermediate representations learned by SRL take full advantage of the rich context in videos. The learned final video representation can be further improved by considering the rich semantic context information by exploring the mid-level semantic tokens, *e.g.*, the activity 'close butter' contains action 'close' and object 'butter'. These tokens can be easily identified from the activity description labels by automatically detecting the nouns and verbs, respectively. Finally, we employ a multi-task learning framework to perform joint feature learning on the targe activity labels and the detected action and object class tokens.

We carry out extensive experiments on two egocentric video datasets (EPIC-KITCHEN and EGTEA Gaze+) to verify the effectiveness of our proposed model. The proposed SRL is also evaluated on third-person video datasets (50 Salads [12] and Breakfast [13]) to prove the generality of our model in predicting future activities. Experiments show that our method achieves promising performances, which sharply outperforms existing state-of-the-art in most cases on the four benchmark datasets. Source codes are available at https://github.com/qzhb/SRL.

The contribution of our work are three folds:

- We propose SRL, a self-regulated learning framework for egocentric activity anticipation. It recursively produces more accurate intermediate representations at any anticipation time-step by iteratively rectifying the current visual representation and reattending to the most relevent observed video frames.
- By exploring mid-level semantic tokens from actions and objects with multi-task learning framework, a more semantically enhanced and self-regulated target representation is obtained for target activity anticipation.
- SRL achieves competitive performance on both egocentric and third-person video datasets. The effectiveness of SRL is also verified by the experimental fact that it can accurately identify action and object concepts that explain the activity semantics.

2 RELATED WORK

We briefly review recent advances in egocentric video analysis in Section 2.1. We further discuss related approaches in third-person video analysis in Section 2.2.

2.1 Egocentric Video Analysis

2.1.1 Egocentric Video Recognition

Egocentric video recognition has been undergoing speedy development in recent years, evidenced by the emergence of many new large-scale egocentric video datasets [1], [2], [3], [4], and a huge body of research work found in literature [2], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23].

As an early endeavor, Spriggs *et al.* [19] utilize a wearable camera and Inertial Measurement Units (IMUs) to explore first-person sensing and perform temporal segmentation

and classification of human activity. Considering the strong contextual relation shown in first-person video, Fathi *et al.* [14] exploit the consistency of appearance representation of actions, hands, and objects, and propose a hierarchical activity modeling framework. Later, it was found that gaze location is a very important clue for egocentric video recognition, so the gaze location is first used for identifying salient visual information in [22]. A probabilistic generative model is proposed to learn the spatio-temporal relationship between gaze point, scene objects, and activity label in first-person video for daily activity recognition in [15].

With the development of deep learning, CNN is employed in [17], [23] for video feature representation in egocentric video recognition. Ma *et al.* [16] design a CNN-based two-stream network to integrate appearance and motion for egocentric activity recognition. Li *et al.* [2] consider the gaze as a probabilistic variable and use a deep network to model its distribution for joint egocentric video recognition and gaze prediction. Sudhakaran *et al.* [20] propose Long Short-Term Attention (LSTA), a new recurrent neural unit to pay attention to features from relevant spatial parts for egocentric activity recognition.

The rapid development of activity recognition deepens our understanding of egocentric video, and also benefits research on other related topics, such as egocentric video summarization, egocentric object detection and egocentric video anticipation.

2.1.2   Egocentric Video Anticipation

Activity anticipation for egocentric vision has been extensively studied in [4], [9], [10], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36]. Some works focus on predicting the next activity. Qi *et al.* [24] propose a spatial-temporal And-Or graph (AOG) to represent events, and an early parsing method using temporal grammar is established to anticipate the next activity. Others focus on predicting what will happen after a long time interval. Ke *et al.* [9] propose to explicitly condition the anticipation on time, which is shown to be efficient and effective for long-term activity anticipation. Furnari *et al.* [30] propose to explore the dynamics of the scene and introduce a model to analyze fixed-length trajectory segments to forecast the next active objects.

Recently, a new egocentric activity anticipation challenge is proposed in [4]. On this challenging dataset, Furnari *et al.* [25] propose new loss functions for activity forecasting, which incorporate the uncertainty of the prediction of future activities. A learning architecture, *i.e.*, RU-LSTM [10], is proposed, which processes RGB, optical flow and object-based features using two LSTMs and a modality attention mechanism to anticipate future activities.

Most of the above activity anticipation techniques have not taken full advantage of the rich context in the video content in their models. Besides, the semantic context information existing among the target activity is left to be uninvestigated. In comparison, we develop a self-regulated learning process which uses the contextual relation along the temporal direction in video sequences on both feature and semantic level to adaptively refine the video features for activity anticipation. Our model can be applied to different

activity anticipation settings, like long time anticipation as in [7] and the challenge in [4] on large-scale datasets.

2.2   Third-person Video Analysis

2.2.1   Third-person Video Recognition

Numerous works based on deep CNN have been proposed for video action recognition. For typical 2D-CNN-based methods [37], 2D convolution is simply applied on single video frame and the frame features are fused. To model temporal information, two-stream-based methods [38], [39], [40] are introduced to model appearance and dynamics separately with two networks and they are fused in the middle stage or at the end. In another branch, many 3D-CNN-based methods (C3D [41], I3D [42], ECO [43] and SlowFast [44]) have been proposed to learn spatio-temporal features from RGB frames directly. Besides, the concept representations are utilized to perform video recognition [45], [46], which is believed to provide better interpretability.

Action recognition models, which can efficiently extract video feature representations, provide backbone models for the development of egocentric video understanding. For example, the TSN model [40] is utilized to extract observed video clip representations for egocentric activity anticipation in RU-LSTM [10]. In our model, the TSN and I3D model are used as video feature extractor.

2.2.2   Third-person Video Anticipation and Prediction

Third-person video anticipation predicts future activity categories from the third-person perspective [7], [11], [47], [48], [49], [50], [51]. Mahmud *et al.* [11] propose a hybrid Siamese network for jointly predicting the label and the starting time of future unobserved activity. Farha *et al.* [7] propose an RNN-based model and a CNN-based model to obtain accurate predictions, which scale well on different datasets and videos with varying lengths and huge variations in the possible future actions.

Third-person action prediction, also referred to as early stage action recognition, aims at predicting the label of an action as early as possible from partial observations [6]. Much effort has been devoted to action prediction [6], [51], [52], [53], [54], [55], [56]. As one of the earliest works on this problem, Ryoo *et al.* [52] use dynamic visual bag-of-words to model changes in feature distribution over time. Shi *et al.* [55] first try to obtain future visual feature through regression, and then an action recognition model is used for action prediction. Kong *et al.* [54] propose a deep sequential context network to reconstruct missing information of the partial observed video for action prediction. Compared to first-person videos, the semantic relation between temporally adjacent video segments appears to be more diversified.

3   APPROACH

Our model SRL is developed upon the general recursive prediction framework, see Figure 2. We first describe the general framework in Section 3.1. Then we show how to perform self-regulated learning in details and discuss how the challenges described in Section 1 are addressed in Section 3.2, Section 3.3 and Section 3.4. Finally, we give our overall learning objective function in Section 3.5.

Fig. 2. The proposed **SRL** framework. SRL consists of three main steps for future activity anticipation. In the observed information encoding step, given the observed video clip I, a feature extractor ϕ and an aggregation function Φ are employed to obtain feature representations F at each observed time-step and hidden representation h_o at the last observed time-step. At the recursive sequence prediction step, a GRU layer is utilized to obtain the initial feature representation h_t^1, then the $Rev.$ loss is employed to rectify it. After that, the revised representation and the observed representation F are fed into the $Rea.$ module to obtain representation f_t^1 that relates to current video content. At last, f_t^1 and h_t^1 are fused by another GRU layer to get the final representation at current time-step. In the target anticipation step, a multi-task learning framework is utilized to enhance the final representation by exploiting the semantic context information (actions $p_{t_s}^v$ and objects $p_{t_s}^n$) related to the target activity, and the predicted activity probability distribution $p_{t_s}^a$ is obtained.

3.1 Future Activity Anticipation Framework

The target of future activity anticipation is to predict the activity label of a video clip starting at time τ_s by observing a video clip starting at $\tau_s - (\tau_o + \tau_a)$ and ending at $\tau_s - \tau_a$, which precedes the target activity start time by τ_a. In other words, we need to predict what will happen after τ_a by observing a video clip of length τ_o. For simplicity, similar as [10], we extract video frames every δ seconds on both the observed part and to-be-anticipated part. Therefore, we assume that the observed video clip contains o frames, represented as $I = \{I_1, I_2, ..., I_o\}$, and the anticipation process contains a frames. We use t to index the current frame to be anticipated and t_s to index the target frame to be anticipated.

As shown in Figure 2, SRL contains three main steps, *i.e.*, the observed information encoding step, the recursive sequence prediction step and the target activity anticipation step. Now we introduce each step in detail.

3.1.1 Observed Information Encoding

For activity anticipation, all the available information we can get for prediction is obtained from the observed video clip. Therefore, how to effectively encode the observed video clip is the foundation for subsequent recursive sequence prediction process. As shown in Figure 2, given the observed video clip $I = \{I_1, I_2, ..., I_o\}$, first a feature extractor ϕ is utilized to obtain the feature representation $F_j \in \mathbb{R}^d$ of the observed video frame at each time-step j. We can use many base models as ϕ, *e.g.*, the TSN model [40] and the I3D model [42]. Then we use an aggregation function Φ (such as RNN model) to encode the observed video representations, and obtain the hidden representation $h_o \in \mathbb{R}^d$ at the last observed time-step. The process is shown as

$$F_j = \phi(I_j); \quad h_o = \Phi(\{F_1, F_2, ..., F_o\}) \tag{1}$$

The obtained representation $F = \{F_1, F_2, ..., F_o\}$ and h_o will be utilized in the following recursive sequence prediction process. The choice of ϕ and Φ will be given in Section 4.2.1 and 4.3.7.

3.1.2 Recursive Sequence Prediction

For recursive sequence prediction, given the observed video representation F and the hidden representation h_o at the last observed time-step, we predict what will happen at the next anticipation time-step repeatedly until the target anticipation time-step is reached.

As mentioned in Section 1, due to error accumulation of recursive sequence prediction, the anticipated intermediate feature representation may be inaccurate. How to utilize the diversified content and rich context contained in the video to regulate the predicted representation is the core issue at the anticipation process for obtaining an accurate and complete intermediate representation.

Specifically, in Figure 2, at each anticipation time-step, given the feature representation h_o at the last observed time-step and the predicted feature representation h_{t-1}^2 at the last anticipation time-step, a GRU layer (GRU_1) is first employed to obtain the initial prediction feature representation h_t^1 at this anticipation time-step as

$$h_t^1 = GRU_1([h_o, h_{t-1}^2], h_{t-1}^1) \tag{2}$$

Applying the initial feature representation directly for subsequent predictions will lead to error accumulation and inaccurate final anticipation results. To get a more accurate feature representation, we utilize the contrastive loss function to rectify the predicted representation. The detail analysis will be shown in Section 3.2.

After the representation revision process, a representation on the video content that is expected to be more accurate at this anticipation time-step is obtained. The next step is to obtain useful information from the observed video clip that is related to the current video content. As shown in Figure 2, the representation h_t^1 will be used to dynamically attend to the observed video representation and acquire

useful information f_t^1 related to h_t^1. Then f_t^1 and h_t^1 are fused to get the final intermediate representation h_t^2, which will be used to anticipate what will happen. We will give detail analysis in Section 3.3.

Finally, we perform the above procedures iteratively until the target anticipation time-step is reached.

3.1.3 Target Activity Anticipation

After the recursive sequence prediction, we can obtain the representation $h_{t_s}^2$ at the final anticipation time-step t_s. Next, we describe in detail how to model semantic context information related to the target activity for obtaining the final accurate activity prediction results in Section 3.4.

The ultimate goal of our model is to get the activity categories at the target anticipation time-step. The probability distribution of the target activity $p_{t_s}^a \in \mathbb{R}^{N_a}$ at the final time-step can be calculated by a linear layer with softmax activation function,

$$p_{t_s}^a = softmax(\boldsymbol{W}_a \hat{h}_{t_s}^2 + b_a) \tag{3}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{d \times N_a}$ is the learnable parameters, and N_a is the number of activity categories. $\hat{h}_{t_s}^2$ is the concatenation of $h_{t_s}^2$ and $h_{t_s}^1$, which increases the representation capability. In our work, we optimize the cross entropy loss L_a to train the activity anticipation model.

3.1.4 Multiple Future Activities Prediction

Once the SRL has been trained, we predict what will happen at multiple future moments following a recursive sequence anticipation style. Given an observed video clip, our model acquires the feature representation of the clip. At each anticipation time-step t, we can get the final feature representation h_t^2 through the Recursive Sequence Prediction module. h_t^2 will be used to obtain the activity label that is taking place at this time-step through the Target Activity Anticipation module. On the other hand, h_t^2 is again forwarded through the Recursive Sequence Prediction module and Target Activity Anticipation module to produce the next prediction. The anticipation results at multiple future time-steps are obtained by repeatedly forwarding the previously generated prediction through the Recursive Sequence Prediction module and Target Activity Anticipation module until the desired final moment is reached.

3.2 Representation Revision with Contrastive Loss

At each anticipation time-step, the obtained initial representation h_t^1 needs to be revised to adapt to the anticipation at time-step t. For a long video, due to the semantic coherence between adjacent video frames, the overall high-level semantic information of a video content should be consistent along the time. For example, the video 'make breakfast' in Figure 1 contains multiple activities. These series of activities are closely related but distinctive. They can be accurately identified by feature representations containing higher-level semantics. Ideally, for an activity anticipation model, the ability to capture event semantics in anticipation stage is necessary for accurate prediction.

Unfortunately, there is no event label on the video subsequence to be anticipated. As an unsupervised learning paradigm, contrastive loss [57] has been widely used in audio and image recognition tasks. It is able to optimize the similarity of sample pairs in the feature space [58], [59] for unsupervised representation learning on high-dimension data.

We apply contrastive loss on our video anticipation task to regulate the predicted feature representation only based on the video content. By using this loss, the representational ability of anticipated features can be enhanced by enforcing the difference between features of different sub-events. A recently proposed contrastive loss function InfoNCE [58] is used in our model. The basic idea is to form a binary classification task that can correctly distinguish the target among a set of samples.

Specially, at each anticipation time-step, given the feature representation set $X = \left\{ x_t^0, ..., x_t^{N-1} \right\}$ with N samples and the initial prediction feature representation h_t^1, the contrastive loss function can be expressed as

$$L_{rev}^t = - \underset{X}{E} \left[\log \frac{\exp(h_t^1 * x_t^0)}{\sum_{x_t^j \in X} \exp(h_t^1 * x_t^j)} \right] \tag{4}$$

where $*$ represents dot product. By minimizing this loss function, we can obtain a revised intermediate feature representation h_t^1.

The set X contains one positive sample that is the feature representation x_t^0 at this anticipation time-step, and $N-1$ randomly sampled negative samples $\left\{ x_t^1, ..., x_t^{N-1} \right\}$. In our method, x_t^0 is obtained by inputting the anticipation frame into the feature extractor ϕ. For the negative samples, to ensure the effectiveness of the representation revision operation, it is expected that the negative samples contain samples with similar (but different) semantic information and with different semantic information to the target sample, where the former can be treated as 'hard negatives'. We split each video into multiple clips according to the activity labels. Each clip is used as a training instance. We sample negative samples randomly from video clips with different activity labels as the selected negative training set. These negative samples for calculating the contrastive loss may come from videos that have the same or different video ids as the positive sample. This setting can better guarantee the diversity and the similarity of the negative samples compared to positive sample. We will give detail analysis on the number of samples N and the sampling methods in Section 4.3.2.

3.3 Dynamically Reattending and Fusion

For activity videos with varied time duration, the contents of key frames have noticeable correlation among one another. For example, in Figure 1(a), the activity 'spread butter' at anticipation time-step t is closely related to the objects 'pancake', 'knife', 'butter' and the action 'open butter' in frames that appear in previously observed video clip. Correspondingly, given a frame to be anticipated, the importance of frames in the observed video clip should be reweighed to enforce those truly related frames. Also, at different anticipation time-steps, the importance of observed frames should be adaptively adjusted due to the content change.

We design a module performing dynamic reattending on the frames in the observed video clip for activity anticipation. Given $h_t^1 \in \mathbb{R}^d$ at anticipation time-step t and the representation $\boldsymbol{F} \in \mathbb{R}^{d \times o}$ of the observed video clip, we define a similarity vector $s_t = \left\{ s_t^1, ..., s_t^j, ..., s_t^o \right\} \in \mathbb{R}^o$, which represents the correlation between the feature at the current time-step and those at each observed time-steps. For example, s_t^j indicates how much useful information we can get from video content at observed time-step j. s_t^j can be calculated as follows,

$$s_t^j = \frac{F_j * h_t^1}{||F_j|| ||h_t^1||} \tag{5}$$

where $*$ represents dot production. Then we use s_t^j to reattend to the useful observed information to get f_t^1 by

$$f_t^1 = \sum_{j=1}^o s_t^j F_j \tag{6}$$

Recall that h_t^1 expresses the video content of the current time-step, and the reattended representation f_t^1 contains more useful relevant information in the past, these two representations can complement each other effectively. To make full use of them, we use another GRU (GRU_2) layer to obtain a more complete feature representation $h_t^2 \in \mathbb{R}^d$ as

$$h_t^2 = GRU_2([h_t^1, f_t^1], h_{t-1}^2) \tag{7}$$

3.4 Semantic Context Exploration

After recursive sequence prediction process with representation revision and reattending, we have obtained h_t^2 that is expected to be more accurate and more comprehensive for target activity anticipation. Besides, for a target activity, there is some useful semantic context information. As shown in Figure 1(a), the activity 'close butter' can be described by mid-level semantics such as action[1] 'close' and object[2] 'butter' that tells the subject and object of the target activity. We refer to these activity-related actions and objects as semantic context. Thus, it is helpful to employ these semantic contexts to make better activity prediction.

Given the main task, *i.e.*, the target activity anticipation by minimizing loss L_a, we construct two auxiliary tasks for action categorization and object categorization by minimizing their respective cross-entropy loss L_v and L_n. The three tasks are performed under the multi-task learning framework, which has been shown to improve the model generality and performance on the main task [60], [61], [62].

Specially, given $h_{t_s}^2 \in \mathbb{R}^d$, we use two separate linear layers with softmax to predict the probability distribution of the related actions and objects,

$$\begin{aligned} p_{t_s}^v &= softmax(\boldsymbol{W}_v \hat{h}_{t_s}^2 + b_v) \\ p_{t_s}^n &= softmax(\boldsymbol{W}_n \hat{h}_{t_s}^2 + b_n) \end{aligned} \tag{8}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{N_v \times N_d}$ and $\boldsymbol{W}_n \in \mathbb{R}^{N_n \times N_d}$ are learnable parameters and N_v and N_n are the numbers of categories of actions and objects, respectively. $\hat{h}_{t_s}^2$ is the concatenation of $h_{t_s}^2$ and $h_{t_s}^1$, which increases the representation capability.

1. The action is described by verb in the activity label.
2. The object is described by the nouns in the activity label.

Together with minimizing L_a, the two cross-entropy loss functions L_v and L_n are minimized to learn the two linear layers.

3.5 Training Objective Function

Given the parameters θ of our model, the overall training objective function can be expressed as,

$$L(\theta) = L_a + \alpha \cdot (L_n + L_v) + \beta \cdot \sum_{t=1}^a L_{rev}^t \tag{9}$$

where $\cdot$ is scale multiplication. $0 \le \alpha \le 1$ and $0 \le \beta \le 1$ are the weights of the corresponding loss function. L_a is an entropy loss for final activity classification, and L_n and L_v are the loss functions for action and object classification of the multi-task learning task. In addition, the third term in Equation 9 is the sum of the contrastive loss at all anticipation time. Our model can be trained end-to-end.

4 EXPERIMENTS

We evaluate SRL on both egocentric and third-person video datasets to verify the general applicability for future activity anticipation.

4.1 Datasets and Metric

4.1.1 Datasets

EPIC-Kitchens Dataset [4] is a large scale cooking video dataset from a first person view captured by 32 subjects in 32 different kitchens. Each video is composed of multiple activity segments, annotated with 125 action and 352 object classes. There are 272 video sequences with 28561 activity segments for training/validation and 160 video sequences with 11003 activity segments for testing. Since the annotations of the test videos are not available, following [10], we split the training set into training and validation sets by randomly choosing 232 videos for training and 40 videos for validation. We consider all unique (action, object) class pairs in the public training set, and obtain 2513 unique activity classes. We also report results on the test set with seen (**S1**) and unseen (**S2**) kitchens. **S1** indicates the test set includes scenes appearing in the training set, and **S2** means the test set includes scenes not appearing in the training set.

EGTEA Gaze+ Dataset [2] contains 28 hours of first person cooking activity videos from 86 unique sessions of 32 subjects performing 7 meal preparation tasks. Each video contains audios, gaze tracking, human annotations of activities and hand masks. This dataset includes 10325 instances of activities, 19 action classes, 51 object classes and 106 unique activity classes. Three different train/test splits are provided by the authors, and we report the average performance of our model across all three splits.

50 Salads Dataset [12] contains 50 videos of salads preparation activities which are performed by 25 actors. The dataset is composed of 17 fine-grained activity classes. As no action and object classes are provided, we decouple 7 unique action classes and 14 object classes from all the activity categories. Following [12], we utilize a five-fold cross-validation for evaluation.

Breakfast Dataset [13] is composed of 1712 videos of people preparing breakfast meals. It contains 48 fine-grained

activity categories. Similar to 50 Salads dataset, we also decouple 15 unique action classes and 36 object classes from all the activity categories. The videos are recorded in 18 different kitchens containing 52 different actors from third person view. The dataset is split into four different train/test splits: S1, S2, S3 and S4. We use these four splits for evaluation.

4.1.2 Metric

For EPIC-Kitchens, following [4], we use the Top-5 accuracy as a class-agnostic measure and Mean Top-5 Recall as a class-aware metric. Specifically, Mean Top-5 Recall is averaged over the provided list of many-shot actions, objects and activities. For the EPIC-Kitchens test set, we use the official evaluation metrics, *i.e.*, Top-1 accuracy, Top-5 accuracy, Average Class Precision and Average Class Recall. For EGTEA Gaze+, Top-5 accuracy is used as the evaluation criterion. For 50 Salads and Breakfast, we use mean accuracy over classes for performance comparison.

4.2 Implementation Details

4.2.1 Experiment Settings

For EPIC-Kitchens and EGTEA Gaze+, all video clips are processed every 0.25s. The input of our model is a fixed-length video clip (*i.e.*, 1.5s in our experiments), and the goal is to anticipate what will happen at multiple time-steps (*i.e.*, 0.25s, 0.5s, 0.75s, 1s, 1.25s, 1.5s, 1.75s and 2s). In other words, the observed time-step o is 6 and the anticipation time-step a is 8. For 50 Salads and Breakfast, we follow the dense anticipation protocol in [7] for the convenience of comparison with other methods. In this setting, the input is a particular percentage (*i.e.*, 20% and 30%) of each video, and the goal is to anticipate the activity labels of the following sub-sequence with a percentage (*i.e.*, 10%, 20%, 30% and 50%) of the video.

For aggregation function Φ, we use a simple GRU layer. Other aggregation function like average pooling can also be utilized in our model. We will give detail analysis of different aggregation functions in Section 4.3.7.

For EPIC-Kitchens and EGTEA Gaze+, we use the feature provided by [10] directly. For 50 Salads, we simply use the feature provided by [63]. For Breakfast, we use I3D [42] to extract the feature representation which will be released along with our source code.

4.2.2 Training Details

For EPIC-Kitchens and EGTEA Gaze+, when we train our model, we first randomly sample a training instance with 14 ($o + a$) frames before the target activity. Then, we split it into 8 training instances with different length of anticipation time-steps (from 1 to 8). Finally, all instances are used to train our model jointly. We use SGD optimizer to train our model. The momentum and weight decay are set to 0.9 and 0.00005, respectively. The mini-batch size is 128. We also utilize dropout layer with dropout ratio 0.5. The provided action and object classes and the synthetic activity classes are utilized as labels to form L_v, L_n and L_a. Specially, for EPIC-Kitchens, we set the initial learning rate as 0.1. The training procedure stops after 100 epochs. For the weight of each loss function, we set α as 0.01 and β as 0.8, and

the setting is determined via a cross-validation. For EGTEA Gaze+, the model is trained with an initial learning rate of 0.1 and 100 epochs. Through cross-validation, we choose α as 0.5 and β as 0.5.

For 50 Salads and Breakfast, considering the dense anticipation protocol, we design a new training instance generation method. Specifically, we enlarge the values of o and a to 16 and 16. Besides, we use a temporal sliding window of 32 ($o + a$) to generate training instances from the beginning to the end of each video. We use Adam optimizer to train our model. β_1, β_2 and weight decay are set to 0.9, 0.999 and 0.00005, respectively. We set the mini-batch size as 128. Dropout layer with dropout ratio 0.5 is also used. We utilize the activity labels to train L_a and the mined action and object labels to train L_v and L_n. For 50 Salads, the learning rate starts from 0.001. The training procedure stops after 100 epochs. We set α as 0.9 and β as 0.1 via cross-validation for the weights of each loss function. For Breakfast, the model is trained with an initial learning rate of 0.01. The training procedure stops after 80 epochs. We choose α as 0.5 and β as 0.5 through cross-validation.

All experiments are implemented under the pytorch framework. For datasets EGTEA Gaze+, Breakfast and 50 Salads, several activity categories contain multiple objects. The annotation template for most of these activity categories is 'put (or place) one object to (or into) another object' (*e.g.*, 'put egg to plate'). For these activity categories, the first object indicates the major object in this activity. Hence, we simply utilize the first object as the object label.

4.3 Ablation Study on EPIC-Kitchens

To analyze the validity of each element in our proposed method, we carry out extensive ablation studies on EPIC-Kitchens and the results are shown in Table 1.

4.3.1 Baseline

For the baseline model ('Baseline' in Table 1), the observed information encoding step is the same as SRL. After that, only a single GRU layer is utilized in the recursive sequence prediction process to predict the feature representation recursively at each anticipation time. Finally, a fully connected layer with softmax activation function is used to predict the target activity. As shown in Table 1, the top-5 accuracy of the baseline is 29.18% at anticipation time 1s.

4.3.2 Baseline+Rev

As shown in Table 1, compared to baseline model, the representation revision operation ('+Rev') boosts the activity anticipation performance from 29.18% to 30.27% at anticipation time 1s, which proves the effectiveness of the representation revision. Actually, at all anticipation times (*i.e.*, 0.25s to 2s), the representation revision operation can lead to performance improvements. By comparing the row of '+Rea & SecCon' and row of 'SRL' in Table 1, we can see that after removing this operation, the performance is degraded to varying degrees at all anticipation times, which further proves the validity of the representation revision operation. The representation revision operation demonstrates even larger improvement with shorter time windows. For example, the performance gap between 'Baseline' and 'Baseline+Rev' at anticipation time 0.5s is 1.26% that is larger

TABLE 1
Ablation studies on EPIC-Kitchens. Given the baseline model, we explore the validity of each component.

Setting	Revision	Reattend	Semantic Context	Top-5 Accuracy % at different τ_a (s)							
				2	1.75	1.5	1.25	1.0	0.75	0.5	0.25
Baseline				24.32	25.06	26.29	27.39	29.18	30.45	31.42	33.75
+Rev	✓			24.42	25.82	27.76	28.64	30.27	31.13	32.68	34.84
+Rea		✓		25.44	26.99	28.22	29.22	30.71	32.30	33.41	35.30
+SecCon			✓	25.46	27.11	27.43	28.96	30.39	31.60	32.64	34.65
+Rev & Rea	✓	✓		25.56	26.81	28.24	29.32	31.23	32.58	33.75	35.32
+Rev & SecCon	✓		✓	25.58	26.77	27.76	28.96	30.89	32.20	33.67	34.98
+Rea & SecCon		✓	✓	25.24	26.29	27.90	28.74	30.77	31.94	33.39	35.38
SRL	✓	✓	✓	**25.82**	**27.21**	**28.52**	**29.81**	**31.68**	**33.11**	**34.75**	**36.89**

TABLE 2
Ablation studies on the sampling methods and the number of samples N on EPIC-Kitchens.

Setting		Top-5 Accuracy % at different τ_a (s)							
Sampling Method	N	2	1.75	1.5	1.25	1.0	0.75	0.5	0.25
same video	128	25.04	26.09	27.53	28.56	30.15	31.05	32.20	34.21
other video	128	25.20	26.47	27.88	28.34	30.37	31.30	32.92	34.77
all video	128	25.26	26.83	28.08	28.94	30.31	31.74	33.23	34.39
all video	32	25.12	26.01	27.90	28.14	30.15	31.92	32.76	35.06
all video	64	25.66	27.01	28.06	28.90	30.33	31.82	33.21	35.40
all video	256	25.42	26.85	27.61	28.14	30.19	31.66	33.15	34.84

than 0.76% at 1.75s. This is mainly because that the feature representation becomes more difficult to be revised as anticipation time increases. Although our model can alleviate the error accumulation, it will inevitably cause a certain degree of error accumulation. Thus, for longer anticipation time, the representation revision operation will face greater challenge compared to situation of shorter anticipation time.

The core of the representation revision operation is the contrastive loss. We sample negative samples randomly from video clips with different activity labels as the selected negative set. We set the value of N as 128. Moreover, we conduct experiments to verify the impact of the samping methods and the number of samples N, as shown in Table 2. 'same video' means we sample negative samples randomly from video clips that have the same video id as the positive sample. 'other video' means we sample negative samples randomly from video clips that have different video id from the positive sample. 'all video' means we sample negative samples randomly from all video clips of the training set.

From the Table 2, we can get several important observations. First, the 'all video' sampling method achieves better results at most anticipation time-steps. However, compared with other sampling methods, the performance advantage is not obvious. On EPIC-Kitchens, videos are captured by different actors in the kitchen scenes. These videos contain diversified and similar semantic information, which leads to more hard negatives in the sampled batch. Hence, sampling from all video clips can better guarantee the diversity and the similarity of the negative samples compared to positive sample, which helps to get more accurate anticipation results. Accordingly, we choose this sampling method in our experiments. Second, the number of samples has a slight influence on our model. Different number of samples has their own prediction performance advantages at some anticipation times. Hence, we choose the appropriate

number of samples according to the convenience of the implementation.

4.3.3 Baseline+Rea

From Table 1, with the dynamically reattending operation ('+Rea' in the table), the performance is 1.53% higher than baseline at anticipation time 1s. The performance improvements are also achieved at other anticipation times, showing that this operation is useful for activity anticipation. Its effectiveness can be further demonstrated by comparing the results in the line '+Rev & SecCon' and line 'SRL' of the Table 1. Besides, by seeing the performance gap between 'Baseline' and 'Baseline+Rea' at anticipation time 0.5s (1.99%) and 1.75s (1.93%), the dynamically reattending operation gives similar improvement for different anticipation times. This phenomenon is different from the representation revision operation. This is mainly because that the dynamically reattending operation can obtain useful information at different anticipation times. At each anticipation time, even though the generated feature representation contains noise, it can still coarsely represent the current video content. Accordingly, the dynamically reattending operation can use the predicted representation to capture useful observed information to some extent. Therefore, the dynamically attending operation is less sensitive to the anticipation time.

4.3.4 Baseline+SecCon

In Table 1, by exploring semantic context information related to the target activity ('+SecCon'), the performance is 1.21% higher than baseline. Moreover, by comparing the results of the line '+Rev & Rea' and the line 'SRL', we can find the performance is degraded at each anticipation time by removing the semantic context exploration operation.

TABLE 3
Egocentric activity anticipation results on the EPIC-Kitchens with different modality features.

Setting		Top-5 Accuracy % at different τ_a (s)							
Mode	Model	2	1.75	1.5	1.25	1.0	0.75	0.5	0.25
RGB	RU [10]	25.44	26.89	28.32	29.42	30.83	32.00	33.31	34.47
	SRL	**25.82**	**27.21**	**28.52**	**29.81**	**31.68**	**33.11**	**34.75**	**36.89**
FLOW	RU [10]	17.38	18.04	18.91	19.97	21.42	22.37	23.49	24.18
	SRL	**17.84**	**18.85**	**19.85**	**20.94**	**21.72**	**23.23**	**24.62**	**25.78**
OBJ	RU [10]	24.54	25.58	26.63	28.32	29.89	30.85	31.82	33.39
	SRL	**25.32**	**26.59**	**27.47**	**28.56**	**30.15**	**31.23**	**33.09**	**34.53**
RGB + OBJ	SRL	29.95	31.19	32.62	34.01	35.32	36.56	38.46	40.12
RGB + FLOW	SRL	26.99	28.06	29.20	30.73	31.94	33.37	35.26	37.47
OBJ + FLOW	SRL	26.93	28.16	29.06	30.43	32.10	33.17	34.59	36.50
Late Fusion	RU(Late) [10]	29.10	29.77	31.72	33.09	34.23	35.28	36.10	37.61
	SRL(Late)	**29.83**	**31.07**	**31.92**	**33.77**	**35.36**	**36.63**	**38.56**	**40.43**
Attention Fusion	RU(Atten) [10]	29.49	30.75	32.24	33.41	35.34	36.34	37.37	39.00
	SRL(Atten)	**30.15**	**31.28**	**32.36**	**34.05**	**35.52**	**36.77**	**38.60**	**40.49**

These phenomenons verify the effectiveness of the semantic context exploration operation.

To show the validity of this operation more clearly, we visualize the semantic context information predicted by SRL in Figure 3. Take the first one as an example, at the target anticipation time, our model obtains the action ('roll') and object ('dough'). Indeed, the 'roll' reveals the action of the target activity and the 'dough' reveals the object involved in the target activity. The obtained action and object do closely relate to the target activity. Hence, the semantic context helps us get more accurate anticipation results. Similar conclusions can also be drawn from other examples.

4.3.5 Baseline+Combination of Any Two Components

The validity of each component of SRL has been demonstrated in the above experiments. We also explore the effectiveness of any two combination of these components, the results are shown in the 5th~7th row in Table 1. We can find that the performance of the combination of two components is higher than that of a single component. For example, the 5th row of Table 1 shows the result of the combination of the representation revision operation and dynamically attending operation, the top-5 accuracy at all anticipation times is higher than any single operation.

4.3.6 On Combining Multiple Modalities

So far, all experiments are based on RGB features. To further verify the validity of SRL, we conduct extensive experiments on other types of feature representations (*i.e.*, optical-flow and object features). The results are shown in Table 3. 'RU(Late)' and 'SRL(Late)' indicate the models using late fusion strategy to merge the prediction results of the three feature modalities. 'RU(Atten)' and 'SRL(Atten)' indicate the models using an attention module to combine the results of the three feature modalities.

For a fair comparison, we use the pre-computed optical-flow and object features provided by [10]. The optical-flow features are extracted using a Batch Normalized Inception CNN. The object features are extracted using Faster R-CNN [64]. See [10] for more details about the utilized features. For models that use optical-flow or object features, the observed time-step o and the anticipation time-step a

are also set to 6 and 8, respectively. A simple GRU layer is used as the aggregation function Φ. We use SGD optimizer with the mini-batch size of 128. For model that uses optical-flow features, the initial learning rate is set as 0.05. The momentum and weight decay are set to 0.9 and 0.00005, respectively. The training procedure stops after 100 epochs. For the weights of each loss function, we set α as 0.5 and β as 0.5, which is determined via cross-validation. For model that uses object features, we train the model with an initial learning rate of 0.1. The momentum and weight decay are set to 0.9 and 0.0001, respectively. The training procedure stops after 100 epochs. Through cross-validation, we set α as 0.8 and β as 0.8 for the loss function.

It can be seen from Table 3 that our model achieves better anticipation performance under different feature modalities (*i.e.*, RGB, OBJ or FLOW) at all anticipation times compared to [10]. Compared to the results using single feature, our model can also achieve higher anticipation accuracy at all anticipation times under the setting of any two feature modalities, *i.e.*, RGB+OBJ, RGB+FLOW, and OBJ+FLOW. This phenomenon suggests that the OBJ and FLOW features are helpful for getting more accurate anticipation results. By comparing the results of RU(Late) and SRL(Late), we can find that our SRL(Late) achieves better performance at all anticipation times using the same features and fusion method. Specifically, at anticipation time 0.25s, our model can improve the top-5 accuracy from 37.61% to 40.21%, resulting in a 2.6% increase. Actually, the abundant visual information and semantic context information contained in the video content is not fully utilized by RU(Late). The better performance of SRL demonstrates the effectiveness and necessity of each component in our model.

Moreover, the performance of SRL(Late) is comparable to that of RU(Atten) at all anticipation time stamps. Note that RU(Atten) designs a Modality ATTention (MATT) module that calculates attention scores to indicate the relative importance of each feature modality for the final anticipation. Therefore, we design a similar attention fusion method to fuse the results of different feature modalities. At the target anticipation time-step, we first concatenate the observed video clip representations of each feature modality. Then, we use an MLP network with three layers

TABLE 4
Results on the EPIC-Kitchens in terms of top-5 accuracy at different anticipation time-steps. 'Act.' means activity.

Model	Top-5 Accuracy % at different τ_a (s)								Top-5 Acc. % @ 1s			M Top-5 Rec. % @ 1s		
	2	1.75	1.5	1.25	1.0	0.75	0.5	0.25	Action	Object	Act.	Action	Object	Act.
DMR [65]	/	/	/	/	16.86	/	/	/	73.66	29.99	16.86	24.50	20.89	03.23
ATSN [4]	/	/	/	/	16.29	/	/	/	77.30	39.93	16.29	33.08	32.77	07.60
MCE [25]	/	/	/	/	26.11	/	/	/	73.35	38.86	26.11	34.62	32.59	06.50
VN-CE [4]	/	/	/	/	17.31	/	/	/	77.67	39.50	17.31	34.05	34.50	07.73
SVM-TOP3 [66]	/	/	/	/	25.42	/	/	/	72.70	38.41	25.42	41.90	34.69	5.32
SVM-TOP5 [66]	/	/	/	/	24.46	/	/	/	69.17	36.66	24.46	40.27	32.69	05.23
VNMCE+T3 [25]	/	/	/	/	25.95	/	/	/	74.05	39.18	25.95	40.17	34.15	05.57
VNMCE+T5 [25]	/	/	/	/	26.01	/	/	/	74.07	39.10	26.01	41.62	35.49	05.78
ED [53]	21.53	22.22	23.20	24.78	25.75	26.69	27.66	29.74	75.46	42.96	25.75	41.77	42.59	10.97
FN [67]	23.47	24.07	24.68	25.66	26.27	26.87	27.88	28.96	74.84	40.87	26.27	35.30	37.77	06.64
RL [68]	**25.95**	26.49	27.15	28.48	29.61	30.81	31.86	32.84	76.79	44.53	29.61	40.80	40.87	10.64
EL [49]	24.68	25.68	26.41	27.35	28.56	30.27	31.50	33.55	75.66	43.72	28.56	38.70	40.32	08.62
RU-RGB [10]	25.44	26.89	28.32	29.42	30.83	32.00	33.31	34.47	/	/	30.83	/	/	/
SRL	25.82	**27.21**	**28.52**	**29.81**	**31.68**	**33.11**	**34.75**	**36.89**	**78.90**	**47.65**	**31.68**	**42.83**	**47.64**	**13.24**

to produce the modality-wise attention score. Finally, the attention score is used to fuse the anticipation results of each feature modality. From the experimental results, we can see that SRL(Atten) can obtain higher performance at all anticipation time stamps compared to RU(Atten) and SRL(Late). In summary, the above experiments verify the validity of single feature and feature combination. For simplicity, in the following experiments, we only use the RGB features in our model.

4.3.7 Different Aggregation Functions

In our model, we consider three crucial factors when we choose GRU as our aggregation function Φ. First, as a sequence model, GRU can encode the observed video information more effectively compared to pooling methods. The aggregated representation h_o at the last observed time-step contains complex history information about the observed video clip. Second, adjacent video frames are more likely to have strong correlation. Using GRU, the aggregated representation h_o can be pushed to the feature representation at the last observation time-step. It assists us to get more accurate prediction results at the first anticipation time-step, and benefit consequent anticipations. Third, compared to LSTM, GRU has fewer parameters. We also try other aggregation functions to compare the experimental results. They are average pooling (Avg), max pooling (Max) and LSTM (LSTM). The results are shown in Table 5. We can find that GRU can get better results compared to other aggregation functions.

4.4 Experiments on Egocentric Video

4.4.1 EPIC-Kitchens

We compare SRL with the following anticipation models: DMR [65], ATSN [4], MCE [25], VN-CE [4], SVM-TOP3 [66], SVM-TOP5 [66], VNMCE+T3 [25], VNMCE+T5 [25], ED [53], FN [67], RL [68], EL [49] and RU-RGB [10]. Note that we only compare RU using RGB features (RU-RGB), and the results using other features are shown in Table 3. We use Top-5 accuracy for activity prediction at different anticipation times (*i.e.*, 0.25s~2s), Top-5 accuracy and mean

TABLE 5
Ablation studies about aggregation function on EPIC-Kitchens.

Setting	Top-5 Accuracy % at different τ_a (s)							
	2	1.75	1.5	1.25	1.0	0.75	0.5	0.25
Avg	**25.93**	27.21	**28.86**	29.67	31.03	32.10	33.02	34.59
Max	26.31	27.13	27.84	29.10	30.61	31.82	32.90	34.13
LSTM	24.96	26.23	27.63	29.00	30.79	31.84	32.98	35.04
GRU	25.82	**27.21**	28.52	**29.81**	**31.68**	**33.11**	**34.75**	**36.89**

Top-5 recall for action, object and activity prediction at anticipation time 1s to evaluate performance. The experimental results are shown in Table 4.

Table 4 clearly shows that SRL achieves the start-of-the-art anticipation performance at most anticipation times. ATSN model, which simply uses the recognition model TSN [40] for activity prediction, only achieves 16.86% Top-5 accuracy at anticipation time 1s. The low performance suggests that we need to design special models to adapt to the video data property of the activity anticipation task. Methods like MCE, FN, RL and EL anticipate the target activity from the observed video clip directly. Instead, methods like RU-RGB and SRL are developed upon the recursive anticipation framework. By comparing the performance differences between these two types of methods, we can find that the recursive anticipation pattern is more suitable for activity anticipation task.

RU-RGB employs a sequence completion pre-training in their model to improve the anticipation performance. Without this pre-training setup, we can still achieve better performance. In fact, there are no enhancements to the predicted feature representation in RU-RGB. The better performance of SRL verifies the necessary of exploiting the informative visual cues contained in the video in the anticipation stage. When inspecting the Top-5 accuracy and the mean Top-5 recall for action, object and activity prediction at anticipation time 1s, we can find a relatively large improvement compared to previous methods. The improvement of action and object prediction performance shows that our semantic context exploration operation is effective, and this indeed

TABLE 6
Results on the EPIC-Kitchens test set with seen (**S1**) and unseen (**S2**) kitchens.

Setting	Model	Top-1 Acc. % @ 1s			Top-5 Acc. % @ 1s			Avg Class Precision. % @ 1s			Avg Class Recall. % @ 1s		
		Action	Object	Act.	Action	Object	Act.	Action	Object	Act.	Action	Object	Act.
S1	2SCNN [4]	29.76	15.15	04.32	76.03	38.56	15.21	13.76	17.19	02.48	07.32	10.72	01.81
	ATSN [4]	31.81	16.22	06.00	76.56	42.15	28.21	23.91	19.13	03.13	09.33	11.93	02.39
	MCE [25]	27.92	16.09	10.76	73.59	39.32	25.28	23.43	17.53	06.05	14.79	11.65	05.11
	RU [10]	33.04	22.78	14.39	79.55	50.95	33.73	25.50	24.12	07.37	**15.73**	19.81	07.66
	TAR [69]	**37.87**	**24.10**	**16.64**	**79.74**	**53.98**	**36.06**	**36.41**	25.20	**09.64**	15.67	**22.01**	**10.05**
	SRL	34.89	22.84	14.24	79.59	52.03	34.61	28.29	**25.69**	06.45	12.19	19.16	06.34
S2	2SCNN [4]	25.23	09.97	02.29	68.66	27.38	09.35	16.37	06.98	00.85	05.80	06.37	01.14
	ATSN [4]	25.30	10.41	02.39	68.32	29.50	06.63	07.63	08.79	00.80	06.06	06.74	01.07
	MCE [25]	21.27	09.90	05.57	63.33	25.50	15.71	10.02	06.88	01.99	07.68	06.61	02.39
	RU [10]	27.01	15.19	08.16	69.55	34.38	21.10	13.69	09.87	03.64	**09.21**	11.97	04.83
	TAR [69]	**29.50**	**16.52**	**10.04**	70.13	**37.83**	**23.42**	20.43	**12.95**	**04.92**	08.03	**12.84**	**06.26**
	SRL	27.42	15.47	08.88	**71.90**	36.80	22.06	20.23	12.48	02.84	07.83	12.25	04.33

TABLE 7
Egocentric activity anticipation results on EGTEA Gaze+.

Model	Top-5 Accuracy % at different τ_a (s)							
	2	1.75	1.5	1.25	1.0	0.75	0.5	0.25
DMR [65]	/	/	/	/	55.70	/	/	/
ATSN [4]	/	/	/	/	40.53	/	/	/
MCE [25]	/	/	/	/	56.29	/	/	/
ED [53]	45.03	46.22	46.86	48.36	50.22	51.86	49.99	49.17
FN [67]	54.06	54.94	56.75	58.34	60.12	62.03	63.96	66.45
RL [68]	55.18	56.31	58.22	60.35	62.56	64.65	67.35	70.42
EL [49]	55.62	57.56	59.77	61.58	64.62	66.89	69.60	72.38
RU [10]	56.82	59.13	61.42	63.53	66.40	68.41	71.84	74.28
SRL	**59.69**	**61.79**	**64.93**	**66.45**	**70.67**	**73.49**	**78.02**	**82.61**

assists us to obtain better activity prediction results.

We also conduct experiments on the EPIC-Kitchens test set with seen (**S1**) and unseen (**S2**) kitchens. The results are shown in Table 6. We can find that SRL obtains better anticipation performance than existing methods except TAR at most evaluation metrics, especially on **S2**. Essentially, TAR creates ensembles of multi-scale feature representations from the observed video clip. This operation is beneficial to predict the next activity. Instead, our SRL addresses the error accumulation issue over long periods of anticipation time. The Top-1 accuracy and Top-5 accuracy metrics are micro-averaged while the Average Class Precision and Average Class Recall metrics are macro-averaged. In a multi-class classification task, the micro-average is preferable if there exists class imbalance. For EPIC-Kitchens, the distribution of categories is imbalance. Accordingly, the higher performance on Top-1 and Top-5 accuracy further verifies the effectiveness of our model.

4.4.2 EGTEA Gaze+

We compare SRL with other anticipation models on EGTEA Gaze+, including DMR [65], ATSN [4], MCE [25], ED [53], FN [67], RL [68]. EL [49] and RU [10]. We evaluate the performance using Top-5 accuracy at different anticipation times (*i.e.*, 0.25s~2s). The comparison results are shown in Table 7. We can find that the performance of SRL is better than all competitors at all anticipation times.

4.5 Qualitative Analysis

To show the anticipation capability of SRL more clearly, we visualize the anticipation results on EPIC-Kitchens in Figure 3. Take the first one as an example. The length of the observed video clip is 1.5s, we can see that the 'dough' is placed on the 'cutting board' step by step and the related 'roll pin' can also be seen in the video. After watching this video, our model can correctly predict the next activity 'rolling dough'. Moreover, in the last example, our model effectively models the information contained in the observed video and accurately predicts the next activity 'put down board'. From the above examples, it can be seen that our SRL can make good use of the observed information and produce accurate prediction.

In order to have a deeper understanding of our model, we also visualize some failure cases in Figure 4. Take the first one as an example, the ground-truth activity is 'move chopstick' and our prediction is 'wipe counter'. From the whole video, we can find that the 'move chopstick' and the 'wipe counter' are two consecutive activities. In the observed video clip, our model can see the activity 'wipe counter'. In the target anticipation time-step, key objects 'chopstick' and 'top' both appear in the scene. Unfortunately, our model predicts the target activity as 'wipe counter' and captures the 'top' as key objects. As we can see from the target frame, the person is moving chopstick with one hand and wiping counter with the other. There is some overlap between the

Fig. 3. The anticipation result visualization. In each example, the observed video clips are shown on the left. The target activity frame and its ground-truth activity category (marked in red) and the predicted activity, action, object category are shown on the right. 'GT' means ground-truth, 'PT' means activity prediction, 'AP' means action prediction and 'OP' means object prediction.

Fig. 4. The failure case visualization. In each example, the observed video frames are shown on the left. The target activity frame and its ground-truth activity category (mark in red) and the predicted activity, action and object category are shown on the right. 'GT' means ground-truth, 'PT' means activity prediction, 'AP' means action prediction and 'OP' means object prediction.

two consecutive activities. Even for activity recognition, this is also a hard example to distinguish. Hence, in this case the anticipation model fails.

4.6 Experiments on Third-person Video

4.6.1 Comparison with Other Methods

In order to verify the generality of SRL, we also conduct experiments on third-person video datasets 50 Salads and Breakfast. We compare SRL with six third-person activity anticipation methods: Nearest-Neighbor, CNN model [7], Grammar-based [70], Uncertainty-based [8], RNN model [7] and Time-cond. [9]. We also compare SRL with the state-of-the-art egocentric video activity anticipaiton method RU-RGB [10]. The experimental results are shown in Table 8.

We can clearly see from Table 8 that SRL outperforms most existing methods on Breakfast. On 50 Salads, in addition to individual prediction moments, SRL also achieves

better activity anticipation results than existing methods. We can also find that the performance of Time-cond. model is better than SRL at some longer anticipation times. This is most likely because the Time-cond. introduces a time parameter t, which denotes the anticipation times. Specifically, the time parameter t is fed to an MLP network to produce a time representation. Then, the time representation and representations of each observed time-step are combined for further processing. This explicit modeling of anticipation time improves the performance of their models for long-term prediction.

We can find poor anticipation performance for 50% anticipation on 50 Salads from Table 8. We think this is mainly due to the unique characteristics of videos in the 50 Salads. First, the length of the video in the 50 Salads varies from more than 4 minutes to more than 10 minutes. Second, there are some background frames that do not contain activity information. These two factors pose great

TABLE 8
Third-person activity anticipation results on 50 Salads and Breakfast. RU-RGB* means our reimplementation of RU [10] using RGB features.

Dataset	50 Salads								Breakfast							
Observed	20 %				30 %				20 %				30 %			
Predicted	10 %	20 %	30 %	50 %	10 %	20 %	30 %	50 %	10 %	20 %	30 %	50 %	10 %	20 %	30 %	50 %
Nearest-Neighbor	19.04	16.10	14.13	10.37	21.63	15.48	13.47	13.90	16.42	15.01	14.47	13.29	19.88	18.64	17.97	16.57
RU-RGB* [10]	22.21	17.81	12.72	08.32	22.30	15.50	10.79	05.18	15.89	14.67	12.46	11.77	15.45	13.55	11.53	10.61
CNN model [7]	21.24	19.03	15.98	09.87	29.14	20.14	17.46	10.86	17.90	16.35	15.37	14.54	22.44	20.12	19.69	18.76
Grammar-based [70]	24.73	22.34	19.76	12.74	29.65	19.18	15.17	13.14	16.60	14.95	13.47	13.42	21.10	18.18	17.46	16.30
Uncertainty-based [8]	24.86	22.37	19.88	12.82	29.10	20.50	15.28	12.31	16.71	15.40	14.47	14.20	20.73	18.27	18.42	16.86
RNN model [7]	30.06	25.43	18.74	13.49	30.77	17.19	14.79	09.77	18.11	17.20	15.94	15.81	21.64	20.02	19.73	19.21
Time-cond. [9]	32.51	27.61	21.26	**15.99**	35.12	**27.05**	**22.05**	**15.59**	18.41	17.21	16.42	15.84	22.75	20.44	19.64	**19.75**
SRL	**37.92**	**28.79**	**21.30**	11.05	**37.46**	24.11	17.05	09.07	**25.57**	**21.04**	**18.54**	**16.03**	**27.31**	**23.59**	**20.83**	17.32

TABLE 9
Ablation studies on 50 Salads. Given the baseline model, we explore the validity of each component.

Setting	Revision	Reattend	Semantic Context	observed 20%				observed 30%			
				10%	20%	30%	50%	10%	20%	30%	50%
Baseline				22.96	18.26	12.96	06.14	22.69	17.12	11.72	06.01
+Rev	✓			27.87	22.34	17.05	09.50	31.42	20.26	13.74	06.97
+Rea		✓		24.08	21.30	15.62	07.57	26.28	18.06	12.75	07.24
+SecCon			✓	24.90	18.36	13.16	07.39	28.08	18.33	13.21	07.32
+Rev & Rea	✓	✓		36.84	26.87	19.59	10.40	33.90	23.75	14.70	07.67
+Rev & SecCon	✓		✓	30.82	23.24	18.14	09.57	36.42	23.05	14.65	08.55
+Rea & SecCon		✓	✓	31.49	26.66	20.80	09.57	32.26	21.06	15.94	08.66
SRL	✓	✓	✓	**37.92**	**28.79**	**21.30**	**11.05**	**37.46**	**24.11**	**17.05**	**09.07**

challenges to the representation revision and dynamically reattending operations in our model. Hence, given a long observed video clip, the performance of SRL for predicting activities over an exceedingly long period may degrade.

Besides, when we focus on the performance differences between RU-RGB and SRL in Table 4 and Table 8, we can find that SRL obtains good anticipation performance on both EPIC-Kitchens and longer-term anticipation on 50 Salads and Breakfast, while RU-RGB is less successful on 50 Salads and Breakfast. This observation indicates the strong predictive performance of SRL on both egocentric and third-person videos.

4.6.2 Ablation Study on 50 Salads

Since the egocentric and third-person activity ancipitation tasks are very different due to the time window they cover (the egocentric anticipation tasks are only focused on the immediate next few seconds rather than the entire rest of the video), we also conduct ablation studies on 50 Salads to see the efficiency of each component of SRL. The results are shown in Table 9.

From the table we can obtain several important conclusions. First, since the third-person datasets do not provide the action and object annotations, we derive the action and object labels from the provided activity annotations. The performance degradation from 'SRL' to '+Rev & Rea' (or from '+SecCon' to 'Baseline') indicates that the prediction of activity-related actions and objects is also necessary for third-person anticipation task. Second, the anticipation performance has different degrees of improvement by adding different single module or any two modules of our SRL to baseline model. These experimental results demonstrate the necessity and the validity of each component of SRL.

Therefore, our specific designed future activity anticipation framework for egocentric videos is also effective for third-person videos. Third, by comparing the ablation study results on egocentric and third-person video datasets in Table 1 and Table 9, we can find that the most effective component of our SRL, which is 'Rev' on 50 Salads and 'Rea' (or 'SecCon') on EPIC-Kitchens, is different for egocentric and third-person ancipitation tasks. Actually, for third-person ancipitation task, it has longer anticipation time window, which may introduce more error accumulation for recursive sequence prediction paradigm. Hence, the 'Rev' module of our SRL will be more significant compared with other components, which can be used to improve the representational ability of the predicted intermediate feature and bring greater performance improvements.

5 WEAKNESS

Our approach performs fairly well in dealing with egocentric video datasets and third-person video datasets, as shown in the experimental results, but there still are several issues to address.

- At each anticipation time, our approach can only give one certain prediction. Since the future is uncertain, like [71], it would be better for our model to produce multiple predictions and give different confidence values.
- When our approach processes long videos with a large number of frames without useful information, the long-term anticipation performance is not as good as the short-term anticipation performance.
- Our sampling methods for representation revision operation cannot avoid sampling certain types of

negative samples. The activity categories of these negative samples are unlikely to co-occur in the same video clip with the activity category of the positive sample, which may provide misleading information and lead to degradation of the training efficacy.

- In order to better solve the error accumulation problem of the recursive sequence prediction paradigm and make full use of the semantic context contained in the video, we need to choose different values of α and β for different datasets and feature modalities, which leads to slight increase of the complexity of our method.

6 CONCLUSION

We have proposed an effective Self-Regulated Learning (SRL) framework to solve the error accumulation problem of recursive sequence prediction pattern for future activity anticipation. SRL aims to regulate the anticipated intermediate representation consecutively to produce more informative representation. Specially, a contrastive loss is utilized to emphasize the novel information in the current anticipation frame in contrast to previously observed content, and a dynamic reweighing mechanism is constructed to exploit the correlation between current frame and previously observed frames, which can attend to informative frames in the observed video clip with a similarity comparison between feature of the current frame and observed frames. Finally, multi-task learning is used to further enhance the learned final video representation, which performs joint feature learning on the target activity labels and the corresponding action and object classes. Experiments on two egocentric video datasets and two third-person video datasets have demonstrated the outstanding performance and effectiveness of the proposed approach. In the future, we will extend our method to other tasks, like the pedestrian trajectory prediction.

REFERENCES

[1] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, "Gaze-enabled egocentric video summarization via constrained submodular maximization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2235–2244.

[2] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 619–635.

[3] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Actor and observer: Joint modeling of first and third-person videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7396–7404.

[4] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.

[5] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 14–29, 2015.

[6] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 269–284.

[7] Y. Abu Farha, A. Richard, and J. Gall, "When will you do what?-anticipating temporal occurrences of activities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5343–5352.

[8] Y. A. Farha and J. Gall, "Uncertainty-aware anticipation of activities," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 1197–1204.

[9] Q. Ke, M. Fritz, and B. Schiele, "Time-conditioned action anticipation in one shot," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9925–9934.

[10] A. Furnari and G. M. Farinella, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6252–6261.

[11] T. Mahmud, M. Hasan, and A. K. Roy-Chowdhury, "Joint prediction of activity labels and starting times in untrimmed videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5773–5782.

[12] S. Stein and S. 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*. ACM, 2013, pp. 729–738.

[13] H. Kuehne, A. Arslan, and T. 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*, 2014, pp. 780–787.

[14] A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding egocentric activities," in *2011 international conference on computer vision*. IEEE, 2011, pp. 407–414.

[15] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in *European Conference on Computer Vision*. Springer, 2012, pp. 314–327.

[16] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1894–1903.

[17] M. S. Ryoo, B. Rothrock, and L. Matthies, "Pooled motion features for first-person videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 896–904.

[18] S. Singh, C. Arora, and C. Jawahar, "First person action recognition using deep learned descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2620–2628.

[19] E. H. Spriggs, F. De La Torre, and M. Hebert, "Temporal segmentation and activity classification from first-person sensing," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2009, pp. 17–24.

[20] S. Sudhakaran, S. Escalera, and O. Lanz, "Lsta: Long short-term attention for egocentric action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9954–9963.

[21] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 229. [Online]. Available: http://bmvc2018.org/contents/papers/0756.pdf

[22] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *CVPR 2011*. IEEE, 2011, pp. 3281–3288.

[23] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora, "Compact cnn for indexing egocentric videos," in *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–9.

[24] S. Qi, S. Huang, P. Wei, and S.-C. Zhu, "Predicting human activities using stochastic grammar," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1164–1172.

[25] A. Furnari, S. Battiato, and G. M. Farinella, "Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation," in *European Conference on Computer Vision*. Springer, 2018, pp. 389–405.

[26] A. Miech, I. Laptev, J. Sivic, H. Wang, L. Torresani, and D. Tran, "Leveraging the present to anticipate the future in videos," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2019, pp. 2915–2922.

[27] T.-Y. Wu, T.-A. Chien, C.-S. Chan, C.-W. Hu, and M. Sun, "Anticipating daily intention using on-wrist motion triggered sensing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 48–56.

[28] S. Z. Bokhari and K. M. Kitani, "Long-term activity forecasting using first-person vision," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 346–360.

[29] C. Fan, J. Lee, and M. S. Ryoo, "Forecasting hands and objects in future frames," in *European Conference on Computer Vision*. Springer, 2018, pp. 124–137.

[30] A. Furnari, S. Battiato, K. Grauman, and G. M. Farinella, "Next-active-object prediction from egocentric videos," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 401–411, 2017.

[31] H. Soo Park, J.-J. Hwang, Y. Niu, and J. Shi, "Egocentric future localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4697–4705.

[32] N. Rhinehart and K. M. Kitani, "First-person activity forecasting with online inverse reinforcement learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3696–3705.

[33] M. Ryoo, T. J. Fuchs, L. Xia, J. K. Aggarwal, and L. Matthies, "Robot-centric activity prediction from first-person videos: What will they do to me?" in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2015, pp. 295–302.

[34] K. K. Singh, K. Fatahalian, and A. A. Efros, "Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[35] B. Soran, A. Farhadi, and L. Shapiro, "Generating notifications for missing actions: Don't forget to turn the lights off!" in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4669–4677.

[36] M. Zhang, K. Teck Ma, J. Hwee Lim, Q. Zhao, and J. Feng, "Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4372–4381.

[37] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[39] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4768–4777.

[40] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[41] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[42] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[43] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 695–712.

[44] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6202–6211.

[45] Z. Qi, S. Wang, C. Su, L. Su, Q. Huang, and Q. Tian, "Towards more explainability: concept knowledge mining network for event recognition," in *The 28th ACM International Conference on Multimedia*. ACM, 2020, pp. 3857–3865.

[46] Z. Qi, S. Wang, C. Su, L. Su, W. Zhang, and Q. Huang, "Modeling temporal concept receptive field dynamically for untrimmed video analysis," in *The 28th ACM International Conference on Multimedia*. ACM, 2020, pp. 3798–3806.

[47] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in *European Conference on Computer Vision*. Springer, 2014, pp. 689–704.

[48] P. Felsen, P. Agrawal, and J. Malik, "What will happen next? forecasting player moves in sports videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3342–3351.

[49] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3118–3125.

[50] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *European Conference on Computer Vision*. Springer, 2012, pp. 201–214.

[51] M. Sadegh Aliakbarian, F. Sadat Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Encouraging lstms to anticipate actions very early," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 280–289.

[52] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1036–1043.

[53] J. Gao, Z. Yang, and R. Nevatia, "RED: reinforced encoder-decoder networks for action anticipation," in *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017. [Online]. Available: https://www.dropbox.com/s/s5yyf1mo8n2f1tq/0284.pdf?dl=1

[54] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1473–1481.

[55] Y. Shi, B. Fernando, and R. Hartley, "Action anticipation with rbf kernelized feature mapping rnn," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 301–317.

[56] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, "Skeleton-based online action prediction using scale selection network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 6, pp. 1453–1467, 2019.

[57] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[58] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: http://arxiv.org/abs/1807.03748

[59] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4182–4192.

[60] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[61] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Advances in Neural Information Processing Systems*, 2018, pp. 527–538.

[62] G. Kapidis, R. Poppe, E. van Dam, L. Noldus, and R. Veltkamp, "Multitask learning to improve egocentric action recognition," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 4396–4405.

[63] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[64] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[65] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 98–106.

[66] L. Berrada, A. Zisserman, and M. P. Kumar, "Smooth loss functions for deep top-k classification," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=Hk5elxbRW

[67] R. De Geest and T. Tuytelaars, "Modeling temporal structure with lstm for online action detection," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*.   IEEE, 2018, pp. 1549–1557.

[68] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1942–1950.

[69] F. Sener, D. Singhania, and A. Yao, "Temporal aggregate representations for long-range video understanding," in *European Conference on Computer Vision*.   Springer, 2020, pp. 154–171.

[70] A. Richard, H. Kuehne, and J. Gall, "Weakly supervised action learning with rnn based fine-to-coarse modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 754–763.

[71] S. Yang, L. Li, S. Wang, D. Meng, Q. Huang, and Q. Tian, "Structured stochastic recurrent network for linguistic video prediction," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 21–29.

Li Su received the Ph.D. degree in computer science from the Graduate University of Chinese Academy of Sciences, Beijing, in 2009. She is currently a Full Professor with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China. Her research interests include image processing and media computing.

Qingming Huang received the B.S. degree in computer science and Ph.D. degree in computer engineering from the Harbin Institute of Technology, Harbin, China, in 1988 and 1994, respectively. He is currently a Chair Professor with the School of Computer Science and Technology, University of Chinese Academy of Sciences. He has authored over 400 academic papers in international journals, such as IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, and top level international conferences, including the ACM Multimedia, ICCV, CVPR, ECCV, VLDB, and IJCAI. He is the Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology and the Associate Editor of Acta Automatica Sinica. His research interests include multimedia computing, image/video processing, pattern recognition, and computer vision.

Zhaobo Qi received the B.S. degree from Harbin Institute of Technology at Weihai in 2016. He is currently pursuing the Ph.D. degree in the School of Computer Science and Technology, University of Chinese Academy of Sciences. His current research interests include video understanding, knowledge engineering and computer vision.

Qi Tian is currently a Chief Scientist in Artificial Intelligence at Cloud BU, Huawei. From 2018-2020, he was the Chief Scientist in Computer Vision at Huawei Noah's Ark Lab. He was also a Full Professor in the Department of Computer Science, the University of Texas at San Antonio (UTSA) from 2002 to 2019. During 2008-2009, he took one-year Faculty Leave at Microsoft Research Asia (MSRA). Dr. Tian received his Ph.D. in ECE from University of Illinois at Urbana-Champaign (UIUC) and received his B.E. in Electronic Engineering from Tsinghua University and M.S. in ECE from Drexel University, respectively. Dr. Tian's research interests include computer vision, multimedia information retrieval and machine learning and published 590+ refereed journal and conference papers. His Google citation is over 26100+ with H-index 78. He was the co-author of best papers including IEEE ICME 2019, ACM CIKM 2018, ACM ICMR 2015, PCM 2013, MMM 2013, ACM ICIMCS 2012, a Top 10% Paper Award in MMSP 2011, a Student Contest Paper in ICASSP 2006, and co-author of a Best Paper/Student Paper Candidate in ACM Multimedia 2019, ICME 2015 and PCM 2007. Dr. Tian research projects are funded by ARO, NSF, DHS, Google, FXPAL, NEC, SALSI, CIAS, Akiira Media Systems, HP, Blippar and UTSA. He received 2017 UTSA President's Distinguished Award for Research Achievement, 2016 UTSA Innovation Award, 2014 Research Achievement Awards from College of Science, UTSA, 2010 Google Faculty Award, and 2010 ACM Service Award. He is the associate editor of IEEE TMM, IEEE TCSVT, ACM TOMM, MMSJ, and in the Editorial Board of Journal of Multimedia (JMM) and Journal of MVA. Dr. Tian is the Guest Editor of IEEE TMM, Journal of CVIU, etc. Dr. Tian is a Fellow of IEEE.

Shuhui Wang received the B.S. degree in electronics engineering from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012. He is currently a Full Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He is also with the Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences. His research interests include image/video understanding/retrieval, cross-media analysis and visual-textual knowledge extraction.

Chi Su is currently a General Manager of Artificial Intelligence Product Center at Kingsoft Cloud, Beijing. He received the PhD degree in the Institute of Digital Media, EECS, Peking University. His research include computer vision and machine learning, with focus on object detection, object tracking, and human identification and recognition.