This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

Egocentric Activity Prediction via Event Modulated Attention

Yang Shen, Bingbing Ni*, Zefan Li, and Ning Zhuang

SJTU-UCLA Joint Center for Machine Perception and Inference, Shanghai Jiao Tong University
{cohakuel, bingbingni, leezf, ningzhuang}@sjtu.edu.cn

Abstract. Predicting future activities from an egocentric viewpoint is of particular interest in assisted living. However, state-of-the-art egocentric activity understanding techniques are mostly NOT capable of predictive tasks, as their synchronous processing architecture performs poorly in either modeling event dependency or pruning temporal redundant features. This work explicitly addresses these issues by proposing an asynchronous gaze-event driven attentive activity prediction network. This network is built on a gaze-event extraction module inspired by the fact that gaze moving in/out of a certain object most probably indicates the occurrence/ending of a certain activity. The extracted gaze events are input to: 1) an asynchronous module which reasons about the temporal dependency between events and 2) a synchronous module which softly attends to informative temporal durations for more compact and discriminative feature extraction. Both modules are seamlessly integrated for collaborative prediction. Extensive experimental results on egocentric activity prediction as well as recognition well demonstrate the effectiveness of the proposed method.

Keywords: Egocentric video, Prediction, Event, Gaze, Attention, Asynchronous

1 Introduction

Egocentric (first-person viewpoint) activity analysis [8, 28, 32] is of particular interest for assisted living. Previous methods [9, 22, 19] mainly focus on activity recognition (*i.e.,* to classify those already occurred activities into different classes); however, for a realistic application, being able to predict an activity before its occurrence is more important, especially in the smart home scenario. For a certain task, the occurrence of activities is usually in order, so modeling the relationship between continuous activities can help to predict the future activity. However, the task of egocentric activity prediction is challenging for most of the existing egocentric methods mainly due to their synchronous processing architecture's limitation in both modeling event dependency and pruning temporal redundant features. On the one hand, the dependency between activities are

* corresponding author: Bingbing Ni.

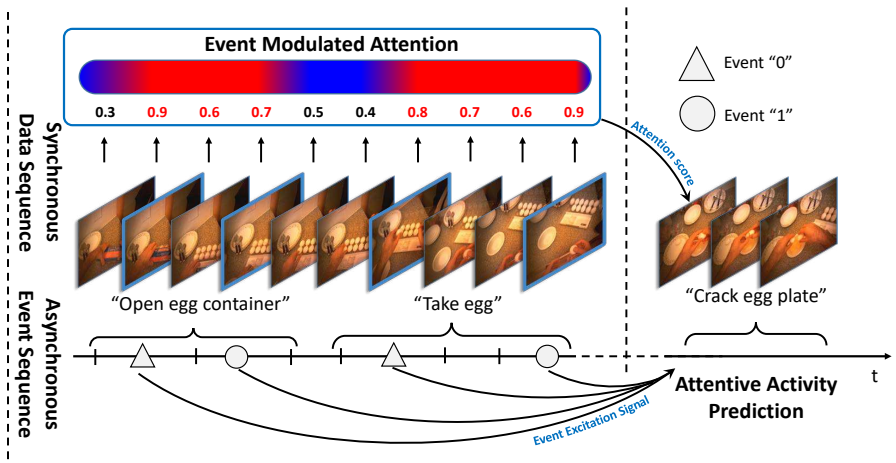


Fig. 1. Motivation overview. Long term asynchronous dependency is of great importance in activity prediction task. Thus we propose a two-branch architecture to deal with both asynchronous event's mutual excitation and synchronous action information. Moreover, each video sequence contains both important and redundant frames. The event modulated attention module is designed to prune redundant features and get a better representation of the sequence.

usually of temporal dependency (and asynchronous). For instance, an incoming activity such as scoop peanut (with) knife might depend on the occurrence of the other activities such as open peanut or "take knife", which might occur several seconds (100 frames) ago. However, current methods such as LSTM-based approaches [2, 29, 4] (*i.e.*, they usually model the dependency no longer than 10 frames) cannot model such a long time dependency. In other words, to predict an egocentric activity, a good model should make use of the previously occurred related events with very long range temporal contexts (i.e., asynchronous dependency). In this paper, inspired by the fact that gaze moving in/out of a certain object closely corresponds to the occurrence/ending of a certain activity, the event is defined as gaze moving in/out of a certain object. On the other hand, most video data recorded by the egocentric camera are redundant which not only convey no useful information for predicting the subsequent event but also induce harmful noise for the task. For example, given a sequence including put cereal, take milk and open milk, the next activity is pour milk (to) bowl, in this case, put cereal has little correlation with the activity to be predicted. So it is redundant and should be omitted. In this sense, a mechanism is required to temporally *attend to* those informative frame features for a higher performance activity predictor.

To explicitly address these issues, this work proposes an asynchronous gaze-event driven attentive activity prediction framework, as illustrated in Fig. 1.

We construct a two-stream asynchronous/synchronous mixed network driven by gaze event. The asynchronous sub-network is constructed based on a Hawkes process model [12], which directly models inter-relationship between different events situated with arbitrary temporal distance. The synchronous sub-network extracts frame-wise deep object and gaze features, and is instantly triggered by gaze events, to output the attended temporal span of informative object/gaze features, yielding discriminative local feature representations for event prediction. Both sub-networks seamlessly collaborate with each other for future activity prediction, and are also trained end-to-end. Extensive experimental results on egocentric activity prediction as well as recognition well demonstrate the effectiveness of the proposed method.

2 Related Work

Egocentric Video Analysis: Currently, egocentric video analysis mainly focuses on activity recognition [8, 28, 32]. CNN was used as an appearance feature extractor in [26, 27], similar to third-person vision activity recognition research. [22] proposed a two-stream network using CNN to analyze appearance and motion information separately. Gaze location is an important cue in egocentric video analysis. Gaze allocation models were usually derived from static picture viewing studies. This has led to methods for computation of image saliency [14] which use low-level image features such as color contrast or motion to provide a good explanation of how humans orient their attention. However, those low-level saliency models performed worse in fixation location prediction compared with those methods based on object-level information [3, 6]. Gaze location was first used as a feature in [9]. Fathi et al. [8] proposed a method modeling the spatio-temporal relationship between gaze, object and activity label by capturing the distribution of visual features and objects in the vicinity of the gaze point. Zhang et al. [31] proposed a generative adversarial neural network based model to anticipate the gaze location beyond the current frame to the future frame.

Event Sequence Analysis: Recurrent neural network [7] was proposed to process sequential data with correlation. Long Short-Term Memory recurrent network (LSTM) is the most successful recurrent neural network architecture which learns the dependency among frames by its special unit structure and it solves the difficulty of training RNN such as explosion and descent vanishing. LSTM was firstly proposed in [13] to comply long-range learning. LSTM was utilized to learn features of 9-frame video clips to realize action classification [2]. LSTM was combined with convolutional neural network (CNN) to further realize video classification [25]. Besides standard time series modeling and prediction of RNN, asynchronous series is also an input to RNN to encode long-range event dependency. Du et al. [5] used the asynchronous event sequence with timestamps about event occurrence as the input to RNNs. Xiao et al. [30] took an RNN perspective to point process, which is an effective mathematical tool to model event data,

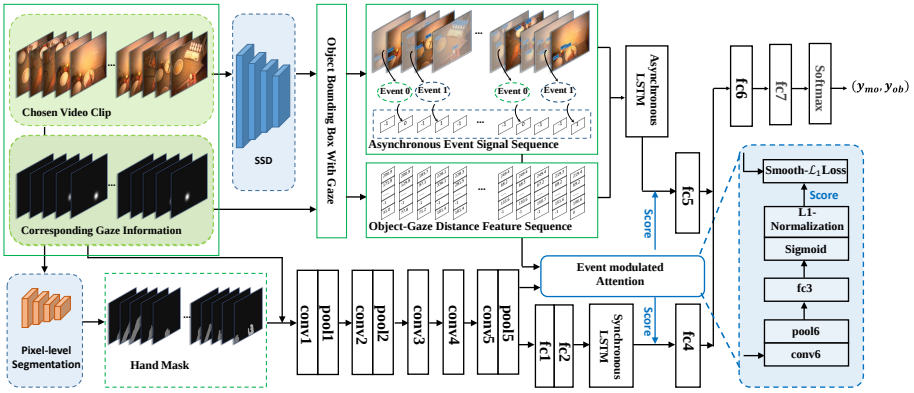


Fig. 2. Overview of our approach. We propose to combine the time-varying features and long-range dependency to predict the future activity. Time series sequence (the lower part) and event sequence (the upper part) can be modeled synergically. Besides, the Temporal Attention Module outputs a sequence of attention scores to decide which frames to attend. Finally, both attentive synchronous and asynchronous features are sent to a softmax classifier to predict the ensuing activity $\hat{\mathbf{y}}_{\mathbf{act}} = (y_{mo}, y_{ob})$.

on the failure prediction for ATMs maintenance. Our work is the first work to integrate both time series and asynchronous series to egocentric video analysis.

Attention Mechanism: The attention mechanism has a great role in sequence learning by adding a model in encoding/decoding section to change the weight of the targeted data. Mnih et al. [23] used an attention-based RNN model to extract information from an image or video only with a sequence of regions selected. Jang et al. [15] proposed a dual-LSTM based method with both spatial and temporal attention, extended Visual Question Answering to the video domain. Liu et al. [21] added the quality score learning to the set-level person re-identification. In this paper, we propose a new event modulated attention triggered by gaze event to deal with the redundant frames.

3 Methodology

State-of-the-art egocentric activity researches, mostly focusing on classification tasks [9, 8, 22, 19], are not capable of predictive tasks, as their synchronous processing architecture performs poorly in modeling event dependency. Another drawback is that, synchronous frames contain lots of redundant information and harmful noise.

Motivated by above limitations, we propose an asynchronous gaze-event driven attentive activity prediction network. More specifically, given a short video clip of N frames $X = \{x_1, x_2, \ldots, x_N\}$, our network predicts the ensuing activity: $\hat{\mathbf{y}}_{\mathbf{act}}$. The architecture of the entire network is illustrated in Fig. 2. The

proposed network extracts both synchronous and asynchronous information. Also, the attention mechanism is applied, to focus on the more informative frame features for a higher performance activity predictor. The whole structure mainly consists of three modules:

- **The Asynchronous Module**, using a gaze-event driven LSTM, taking sequences of event data as the input triggered by gaze, deals with temporal dependency between events with arbitrary distance.
- **The Synchronous Module**, using a time series LSTM, taking the hand mask and gaze location information as the input, deals with synchronous frame information, i.e., instant feature-event relationship.
- **Event Modulated Attention**, designed as a convolutional network, learns soft attention scores to temporally attend to those informative frame features.

Then a softmax classifier is applied to fuse the extracted synchronous and asynchronous features to predict the ensuring activity (right behind the given video clip): $\hat{\mathbf{y}}_{\mathbf{act}}$. Here the activity is defined as motion + objects (e.g., "crack"+"egg"). $\hat{\mathbf{y}}_{\mathbf{act}} = (y_{mo}, y_{ob})$, y_{mo} and y_{ob} represent the motion and object label, respectively.

3.1 Asynchronous Module

To model event dependency, the asynchronous module is constructed based on Hawkes process [12], which directly models inter-relationship between different events situated with arbitrary temporal distance. Hawkes process is a type of point process. Point process is a principled framework for modeling event data [1] and interdependency between events, which lies with arbitrary distance along the temporal axis. The conditional intensity function is originally defined as follow:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{E(N(t+\Delta t) - N(t)|H_t)}{\Delta t} = \frac{E(dN(t)|H_t)}{dt}, \tag{1}$$

where $\lambda(t)$ is the rate of the occurrence of a new event conditioned on the history H_t, for a short time interval $[t, t+dt)$. $E(dN(t)|H_t)$ represents the expectation of the number of events happened in the interval $[t, t+dt)$ given the historical observations H_t.

In Hawkes process, the conditional intensity function is defined by a specific parameterization:

$$\lambda_{Hawkes}(t) = \mu(t) + \sum_{t_i < t} \delta(t, t_i), \tag{2}$$

where $\delta(t, t_i)$ is the time-decaying kernel. $\mu(t)$ represents the background effect. $\sum_{t_i<t} \delta(t, t_i)$ is the excitation effect from history events, modeled by a trigger term. Hawkes process can help to model the excitation relationship of the happened events and the coming events, which is important for our prediction task.

Gaze information is significant in egocentric video analysis, for eyes usually lead to the next activity before the hands. Gaze movement is an important

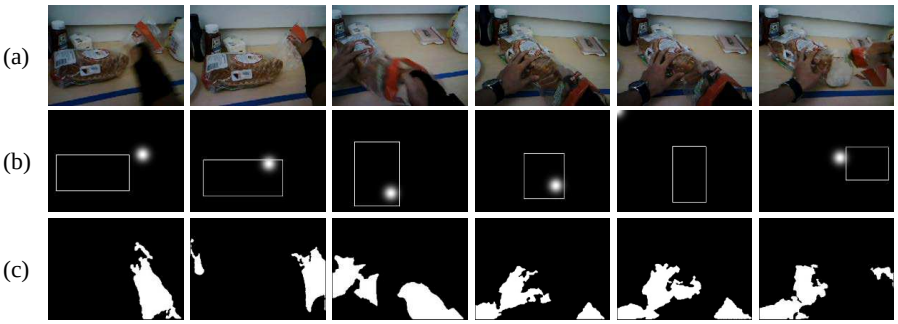


Fig. 3. Data examples of GTEA Gaze dataset. (a) Original video images; (b) the relationship between gaze movement and the object being manipulated, gaze moving in/out is signed as '0/1'; (c) the hand masks we get from the pixel-level segmentation network [33].

cue for the ensuing activity analysis. Thus, we define the asynchronous events triggered by gaze (examples can be seen in Fig. 3):

- Event '**0**' occurs when the gaze point moves into the target object in a certain frame of the video clip.
- Event '**1**' occurs when the gaze point leaves the target object in a certain frame of the video clip.

As can be seen in the upper part of Fig. 2, these two types of events are captured to get an asynchronous event signal sequence. To capture these asynchronous events, following steps are adopted:

Firstly, for an egocentric video dataset, a single shot multibox detector (SSD) [20] is trained to give bounding boxes for all the objects in the dataset. Secondly, sliding windows are used to incept small video clips from the whole dataset, with each clip consisting of $90 \sim 120$ frames. Clips contain too few or too many events should be removed because too few events cannot provide enough information for prediction while those containing too many events indicates the excessively high moving frequency of the gaze point, which also has a negative influence in prediction. The rest video clips constitute the training sets. For each video clip, the detected object bounding box information is combined with the corresponding gaze information to generate the asynchronous event signal sequence:

$$Z = \{z_i\}_{i=1}^{N} = \{z_1, z_2, \ldots, z_N\}, \tag{3}$$

where $z_i \in \{-1, 0, 1\}$ denotes the event type in the i-th frame:

$$z_i = \begin{cases} 0 & \text{event '0' occurs in the i-th frame;} \\ 1 & \text{event '1' occurs in the i-th frame;} \\ -1 & \text{neither event '0' or '1' occurs.} \end{cases} \tag{4}$$

Suppose there are precisely m types objects in the dataset, labeled from 1 to m. For the i-th frame in a video clip, with object and gaze-point information, we define the object-gaze distance feature sequence:

$$\boldsymbol{d_i} = \left[d_{i1}, d_{i2}, ..., d_{im}\right]^T, \tag{5}$$

where d_i denotes the euclidean distance between the gaze point and the center of the i-th object. $d_i = -1$ when the i-th object fails to be detected in this frame.

For a chosen video clip with N frames: $X = \{x_1, x_2, ..., x_N\}$, the corresponding input sequence X^a for the asynchronous module is:

$$X^a = \{\boldsymbol{x_i^a}\}_{i=1}^N, \tag{6}$$

where $\boldsymbol{x_i^a} = \begin{bmatrix} z_i \\ \boldsymbol{d_i} \end{bmatrix} = \left[z_i, d_{i1}, d_{i2}, ..., d_{im}\right]^T$.

3.2 Synchronous Sequence Module

As shown in Eq. 2, besides history event excitation, the background excitation $\mu(t)$ is also an important cue for event modeling. In our method, the background excitation is modeled by a time series LSTM (the synchronous module). The lower part of Fig. 2 detailedly describes the structure of the synchronous module.

For each frame x_i in a chosen video clip ($X = \{x_1, x_2, ..., x_N\}$), the hand mask (denoted as H_i) and gaze point (denoted as G_i) information are encoded as the input features for the synchronous module. The hand mask H_i is extracted by a pixel-level segmentation network [33], which adopts a low resolution FCN32-s and uses the sum of per-pixel two-class softmax losses as the loss function. The gaze-point information is a 2D coordinate which denotes the gaze location in the original frames. To enhance the gaze information, we map a normal distribution (mean value $\mu = 0$ and variance $\sigma^2 = 0.2$) to the gaze point, and get a gaze-point map G_i. The input sequence X^s for the synchronous module is:

$$X^s = \{x_i^s\}_{i=1}^N, \tag{7}$$

where $x_i^s = H_i \oplus G_i$, $\oplus$ denotes the concatenation operation along channel axis.

3.3 Temporal Attention for Two-Stream LSTM

Event Modulated Attention: For a video clip, some frames may have little correlation with the activity to be predicted, which include harmful noise and should be omitted. Our hypothesis is that frames between event '0' and event '1' deserve more attention (frames start from event '**0**' and end in event '**1**'). For example, as shown in Fig. 1, the asynchronous event signal sequence is $Z = \{-1, 0, -1, 1, -1, 0, -1, -1, 1, -1\}$. We can generate the binary attention mask: $M = \{0, 1, 1, 1, 0, 1, 1, 1, 1, 0\}$, which highlights the useful information (signed as '1') and drops the useless frames (signed as '0').

However, directly applying binary attention mask leads to numerical problems in latter LSTM cells. Thus we propose **Event Modulated Attention** (in Fig. 2) to learn a soft attention score. This module is a convolutional network with input features extracted from the pool5 layer of AlexNet. The output spatial is $256 \times 6 \times 6$. The convolution part contains a convolutional layer with kernel size 3×3, stride 1, 256 output-channels and a mean-pooling layer with the stride of 2 and kernel size of 2×2. Then a fully connected layer follows to generate a $N \times 1$ raw score $\tilde{\boldsymbol{q}} = [\tilde{q_1}, \tilde{q_2}, \ldots, \tilde{q_N}]$. $\tilde{\boldsymbol{q}}$ is activated and normalized by a sigmoid layer and a group L1-normalization layer to get the final attention score $\boldsymbol{q} = [q_1, q_2, \ldots, q_N]$. We use the binary attention mask M generated from the corresponding asynchronous event signal sequence Z as the supervised signal for this modular. To avoid gradient vanishing problem of $L1$-Loss, we apply the smooth $L1$-Loss [10]:

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| & |x| \geqslant 1. \end{cases} \tag{8}$$

Finally, the attention score $\boldsymbol{q}$ functions as a feature score to determine the importance of different frame features.

LSTM block: LSTM [13] is a powerful tool in dealing with the sequential input. Having input sequence: $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}\}$, LSTM generates the hidden states $\{\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_N}\}$ and output a sequence [7]. A basic LSTM block include three gates, the input gate $\boldsymbol{i_t}$, the forget gate $\boldsymbol{f_t}$ and the output gate $\boldsymbol{o_t}$, it updates as follows [11]:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f) \\ c_t &= f_c c_{t-1} + i_t * tanh(W_c x_t + U_c h_{t-1} + b_c) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + V_o c_{t-1} + b_o) \\ h_t &= o_t * tanh(c_t) \end{aligned} \tag{9}$$

where c_t is a single memory cell. σ means sigmoid function and $*$ represents the element-wise multiplication operator. $\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{V}$ are the weighted matrices and $\boldsymbol{b}$ is the bias vector. $\boldsymbol{x_t}$ and $\boldsymbol{h_t}$ represent the input feature vector and the hidden output vector. The update equation of $\boldsymbol{c_t}$ is composed of two parts: a fraction of the previous cell state $\boldsymbol{c_{t-1}}$ and a new input state created.

Two-stream LSTM: In the proposed framework, we design two individual LSTM modules: the synchronous module, with its units aligned with the timestamps of a time series, and the asynchronous module, whose units are aligned with events. As shown in Fig. 2, two LSTM modules are designed as follow:

- To capture the long-range dependency over history with arbitrary time intervals, the asynchronous part takes the object-gaze distance and event signal as its input.
- The synchronous part takes the hand mask and gaze-point information as its input and is designed to timely track the temporal information.

Two fully connected layers are established after LSTM. The whole network is supervised by a Softmax Loss:

$$L_{class} = \frac{1}{N}\sum_{i=1}^{N} -\boldsymbol{y_i}\log\hat{\boldsymbol{y}}_{\boldsymbol{i}} - (1-\boldsymbol{y_i})\log(1-\hat{\boldsymbol{y}}_{\boldsymbol{i}}), \tag{10}$$

where N is the number of training samples, $\boldsymbol{y_i}$ represents the ground truth and $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ is our predicted label.

4 Experiments

In this section we briefly introduce the datasets (Section 4.1), then analyze the temporal dependency between activities (Section 4.2) and present experimental results of three tasks of activity prediction (Section 4.3), recognition (Section 4.4) and robustness analysis (Section 4.5).

4.1 Datasets

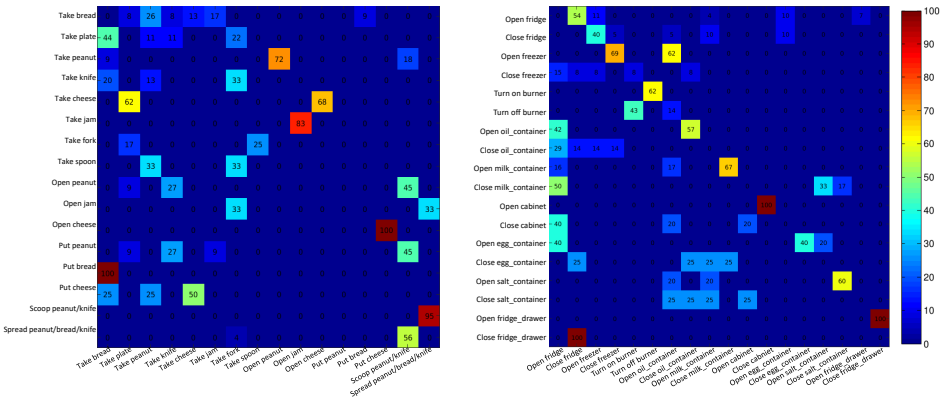
In our work, we use two public datasets: GTEA Gaze [9] and GTEA Gaze+ [8]: Both of them contain the subjects' gaze location in each frame and the activity labels.

- GTEA Gaze (Gaze): This dataset contains 17 sequences of meal preparation activities performed by 14 different subjects, with the resolution of 640×480.
- GTEA Gaze+ (Gaze+): This dataset contains 37 sequences performed by 6 subjects of preparing 7 types of meals, with a higher resolution of 960×720.

4.2 Temporal Dependency Between Activities

We extend the typical egocentric activity recognition task to a future activity prediction task, for there exists strong relevance between the neighboring activities (for example, after the activities take milk and open milk, there is a great possibility that pour milk will happen).

To statistically analyze the temporal dependency between neighbouring activities, we collect 6 sequences of making north American breakfast in Gaze+ and 5 sequences of making sandwich in Gaze. Neighboring activity distribution is shown in Fig. 4, the vertical coordinate denotes the current activity and the horizontal coordinate denotes the next activity. Each row of this matrix represents the occurrence probability percentage of the next activity after the current activity. Our hypothesis is that there exists temporal dependency between neighbouring activities. To verify this, we apply the Spearman Correlation Analysis. The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the ranked variables, and it is appropriate for both continuous and



(a) (b)

Fig. 4. Statistics on neighboring activities, best viewed in color. (a) Gaze 16 classes; (b) Gaze+ 18 classes. The vertical coordinate denotes the current activity and the horizontal coordinate denotes the next activity. Each row of this matrix represents the occurrence probability percentage of the next activity after the current activity.

discrete ordinal variables [18]. The Spearman correlation coefficient is computed as follows:

$$\rho = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 (y_i - \overline{y})^2}}, i = 1, 2, ..., N, \tag{11}$$

where x_i and y_i are the original data, $\overline{x}$ and $\overline{y}$ are the mathematical expectation. The Spearman correlation coefficient for Gaze+ is 0.43 and the corresponding p-value is $6.97 \times 10^{-7} \ll 0.05$. According to Hypothesis Testing theory, we can strongly believe that there exists moderate dependency between neighboring activities. Thus it is reasonable for us to model the dependency between neighbouring activities to predict the future activity.

4.3 Activity Prediction

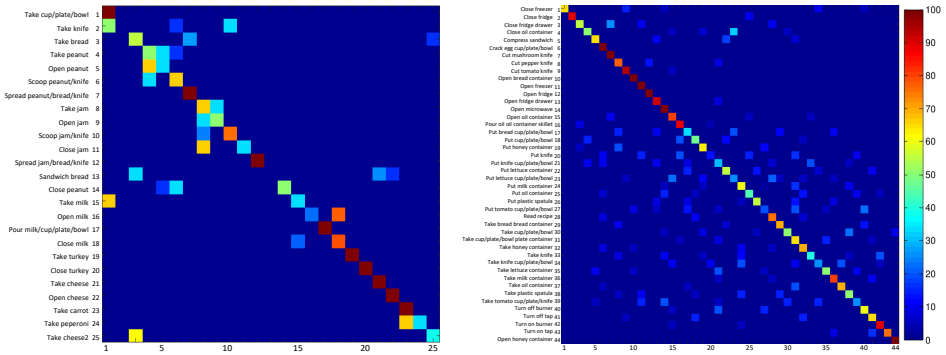
We use 13 sequences for training and 4 sequences for test on GTEA Gaze, 30 sequences for training and 7 sequences for test on GTEA Gaze+. The test set includes each type of meal preparation. As we discussed in Section 3.1, we use sliding windows to extract small video clips (1528 for Gaze, 4151 for Gaze+) as our training samples, with each containing $90 \sim 120$ frames. Also, we get the hand mask, event signal sequence and object-gaze distance feature sequence during data preparation. The training stage includes the following steps: i) train the synchronous branch separately (Time series LSTM, lower part in Fig. 2, with pre-trained AlexNet [17].) without attention scores and asynchronous features. ii) train the asynchronous branch separately (Event sequence LSTM, upper part in Fig. 2) without attention scores and synchronous features. iii) train the whole network with attention module and both branches.

Table 1. Performance for activity prediction and recognition. (a) Results from Fathi et al. [8] using the observed gaze; (b) Two-stream CNN results with object-cnn, SVM-fusion and joint training [22]; (c) 2D and 3D Ego ConvNet results (**H**: Hand mask, **C**: Camera/Head motion, **M**: Saliency map) [28]. (d) Results of our method, for activity recognition, we use the adjusted network with two synchronous models. Gaze (RB) and Gaze+ (RB) represent the sub-datasets re-annotated by the Rubicon Boundaries labeling method.

	Methods	Prediction		Recognition			
		Gaze	**Gaze+**	**Gaze**	**Gaze+**	**Gaze(RB)**	**Gaze+(RB)**
[8]	observed gaze	-	-	0.47	0.51	0.48	0.52
[22]	object-cnn	0.442	0.438	0.471	0.464	0.487	0.473
	motion+object-svm	0.192	0.264	0.284	0.347	0.305	0.352
	motion+object-joint	0.576	0.601	**0.624**	0.664	0.636	0.668
[28]	H+C+M(2D)	0.437	0.462	0.508	0.534	0.523	0.538
	H+C+M(3D)	0.492	0.504	0.525	0.542	0.536	0.553
	H+C+M(2D+3D)	0.514	0.537	0.549	0.581	0.560	0.589
Ours	Time series LSTM	0.581	0.614	0.619	**0.671**	**0.654**	**0.686**
	Event sequence LSTM	0.612	0.659	-	-	-	-
	Fusion LSTMs	0.632	0.674	-	-	-	-
	Attention based LSTMs	**0.648**	**0.687**	-	-	-	-

For each state, we use the same training strategy: stochastic gradient descent with momentum=0.9, weight decay=0.0005. We apply exponential decay to learning rate, with initial learning rate 0.0001 for Alexnet and 0.001 for two LSTM modules. We conduct our experiments on the open source Caffe framework [16]. For prediction baselines, most related works focus on the activity recognition task. Thus we adjust two state-of-the-art works [22, 28] to activity prediction task, with each containing three different methods. To do so, we simply replace the recognition label with the prediction label. For our own methods, we test four different network versions as follows: **1) Time series LSTM:** without attention and asynchronous information; **2) Event sequence LSTM:** without attention and synchronous information; **3) Fusion LSTMs:** concatenating both asynchronous and synchronous features; **4) Attention based LSTMs:** concatenating both asynchronous and synchronous features with soft attention scores. The reproduced experiment results are shown in the prediction part of Table 1.

The event sequence LSTM outperforms time series LSTM, which suggests that history event effects are important for future activity occurrence. The proposed two-stream LSTM without attention outperforms [22] and [28] by 5.6% (7.3%) and 11.8% (13.7%) on Gaze (Gaze+) respectively. The reason for this improvement is that previous methods only utilize synchronous information while our network makes use of event triggered asynchronous information. Moreover, event modulated attention enhances the prediction accuracy by 1.6% and



(a) (b)

Fig. 5. Confusion matrix of our proposed method for activity prediction, best viewed in color. (a) Gaze 25 classes; (b) Gaze+ 44 classes.

1.3% on Gaze and Gaze+ respectively. This is because the temporal attention mechanism largely reduces negative impact of redundant and noise frames. The confusion matrices (using two-stream LSTM with attention) are shown in Fig. 5.

To further show the importance of gaze movement to the activity, we also test the accuracy of single motion prediction. Shown in the left part of Table 2, our results outperform our baseline [22]. The reason is that the sequence of the gaze movement information is the most important cue for motion prediction. The traditional method [22] to use optical flow/CNN to analyze motion is easily influenced by the camera and subjects' shake, while our attention mechanism can solve the problem.

4.4 Activity Recognition

We apply our prediction framework to a set level activity recognition task. We extract new video clips (3568 for Gaze, 10624 for Gaze+) as our training samples, with each video clip containing 7 frames of the same label. We adjust the asynchronous branch to another synchronous branch by removing the event signal sequence, with the original synchronous branch remaining the same. Thus our activity recognition network (containing two synchronous branches) consists of two time series LSTM modules. The train strategy is similar to activity prediction task.

For contrast experiments, we train three different methods on Gaze and Gaze+ [8, 22, 28]. Observed gaze method is adopted by Faith et al.[8], modeling the spatio-temporal relationship between gaze, object and activity label by capturing the distribution of visual features and objects in the vicinity of the gaze point. The other two models [22, 28] achieve state-of-the-art results, which are our baselines. Results are shown in the recognition part of Table 1. Our method outperforms the state-of-the-art methods in Gaze+ and is slightly inferior to the

Table 2. Performance of motion prediction and recognition. (a) Two-stream CNN results of joint training from Ma et al. [22]; (b) Results of our method, attention based LSTMs for motion prediction and time series LSTM for motion recognition.

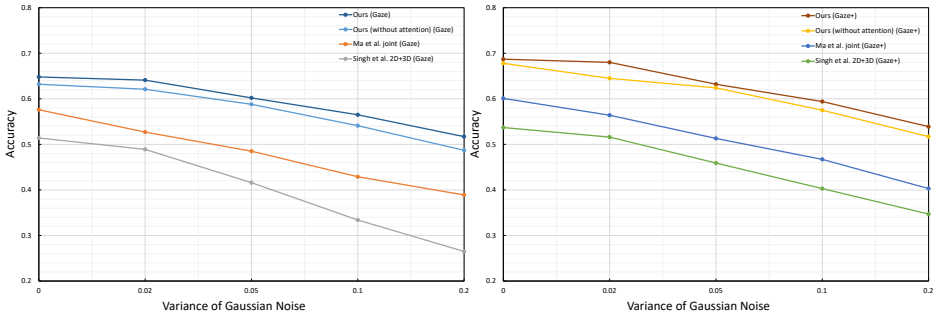
	Methods	Prediction		Recognition	
		Gaze	**Gaze+**	**Gaze**	**Gaze+**
[22]	Joint training CNNs	0.308	0.576	0.363	0.651
Ours	Time series LSTM	-	-	**0.526**	**0.788**
	Attention based LSTMs	**0.612**	**0.842**	-	-

joint training method of [22] in Gaze. One reason is that our method is set level recognition while baselines are all frame level recognition. Frames in a sequence are complementary and using the features extracted from the frame sequence can hopefully lead to higher accuracy in action recognition. Another reason is that Gaze and Gaze+ contain many transition frames (between neighboring activities), resulting ambiguous labeling problems among these frames. Thus we use the Rubicon Boundaries labeling method proposed by [24] to re-annotate the labels of Gaze and Gaze+ (denoted as Gaze (RB) and Gaze+ (RB)). We only use the sub-segment of activity phases as our sub-dataset and drop the sub-segment of pre-activity and concatenation phases. Results are shown in Table 1. Our method outperforms all other methods by a large margin. We also test the accuracy of motion recognition, our method outperforms the baseline [22], which shows the gaze movement can bring more information of motion than optical flow, because optical flow to analyze motion is easily influence by the camera shake.

4.5 Robustness Analysis

To test the robustness of our network, we randomly add Gaussian noise with different variances on the features before they are sent into LSTM on the activity prediction task (using two-stream LSTM with/without attention mechanism). For the synchronous module, we randomly add noise on the concatenation of hand mask and gaze. For the asynchronous module, we add noise on the bounding box scores after object localization network. For our baselines, we add the same random noise on the hand mask, saliency map and optical flow.

Results from Fig. 6 show that our methods outperform our baselines after adding Gaussian noise of different variances. Accuracy of our two-stream LSTM without attention drops 14.5% (15.5%) on Gaze (Gaze+), while the declines are 19.7% (19.8%) of Ma et al. [22] and 24.9% (21.0%) of Singh et al. [28] on Gaze (Gaze+). We conclude that it is mainly due to different feature representations. Our methods use sequence information as the input and mainly focus on long-term context features which are not sensitive to the single frame noise, while our baselines focus on frame level recognition, more sensitive to single frame noise. The declines are 13.1% (13.9%) on Gaze (Gaze+) after adding the event



(a) (b)

Fig. 6. Results of activity prediction by adding Gaussian noise on our method and baselines of both datasets (a) Gaze; (b) Gaze+, best viewed in color. The dashed and solid lines are results on Gaze and Gaze+ respectively. The methods we show are fusion LSTMs of our method, motion-object joint training of Ma et al. [22] and Ego ConvNet with 2D and 3D of Singh et al. [28]. Note that noise has the least effect on our method.

modulated attention module on two-stream LSTM, which shows that the soft attention scores we obtain from the temporal attention module can further reduce the impact of single frame noise. That is because the temporal attention module can attend to those frames that are more important and our baselines take all the frames equally.

5 Conclusion

We extend the typical egocentric activity recognition task to a future activity prediction task, as we prove that there exists moderate relevance between the neighboring activities. We have developed a gaze-event driven attentive activity prediction network to integrate both synchronous and asynchronous information, modeled as background and event excitation. The asynchronous event is defined as gaze moving in/out of the manipulated . We believe that our work will certainly help advance the field of egocentric activity analysis.

6 Acknowledge

This work was supported by National Science Foundation of China (U161146161502301, 61521062). This work was supported by SJTU-UCLA Joint Center for Machine Perception and Inference.The work was also partially supported by Chinas Thousand Youth Talents Plan, STCSM 18DZ2270700 and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China. This work was supported in part by NSFC 61671298, STCSM 17511105400.

References

1. Aalen, O., Borgan, O., Gjessing, H.: Survival and event history analysis: a process point of view. Springer Science and Business Media (2008), https://doi.org/10.1162/neco.1997.9.8.1735
2. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Action classification in soccer videos with long short-term memory recurrent neural networks. In: ICANN. pp. 154–159 (2010)
3. Borji, A., Sihite, D.N., Itti, L.: Probabilistic learning of task-specific visual attention. In: CVPR. pp. 470–477 (2012)
4. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. In: Proceedings of SSST@EMNLP. pp. 103–111 (2014), http://aclweb.org/anthology/W/W14/W14-4012.pdf
5. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD International Conference. pp. 1555–1564 (2016)
6. Einhauser, W., Spain, M., Perona, P.: Objects predict fixations better than early saliency. Journal of Vision **8**(14), 18.1 (2008)
7. Elman, J.L.: Finding structure in time. Cognitive Science **14**(2), 179–211 (1990)
8. Fathi, A., Li, Y., Rehg, J.M.: Learning to recognize daily actions using gaze. In: ECCV. pp. 314–327 (2012)
9. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: CVPR. pp. 3281–3288 (2011)
10. Girshick, R.B.: Fast R-CNN. CoRR **abs/1504.08083** (2015), http://arxiv.org/abs/1504.08083
11. Graves, A.: Generating sequences with recurrent neural networks. CoRR **abs/1308.0850** (2013), http://arxiv.org/abs/1308.0850
12. Hawkes, G., A.: Spectra of some self-exciting and mutually exciting point processes. Springer Science and Business Media (1971), https://doi:10.2307/2334319
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
14. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
15. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: TGIF-QA: toward spatio-temporal reasoning in visual question answering. In: CVPR. pp. 1359–1367 (2017)
16. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACMMM. pp. 675–678 (2014), http://doi.acm.org/10.1145/2647868.2654889
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1106–1114 (2012)
18. Lehman, A., O'Rourke, N., Hatcher, L., stepanski, E.: Jmp for basic univariate and multivariate statistics: Methods for researchers and social scientists second edition p. 123 (2005)
19. Li, Y., Ye, Z., Rehg, J.M.: Delving into egocentric actions. In: CVPR. pp. 287–295 (2015)
20. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: ECCV. pp. 21–37 (2016)

21. Liu, Y., Yan, J., Ouyang, W.: Quality aware network for set to set recognition. In: CVPR. pp. 4694–4703 (2017)
22. Ma, M., Fan, H., Kitani, K.M.: Going deeper into first-person activity recognition. In: CVPR. pp. 1894–1903 (2016)
23. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. In: NIPS. pp. 2204–2212 (2014), http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention
24. Moltisanti, D., Wray, M., Mayol-Cuevas, W.W., Damen, D.: Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In: ICCV. pp. 2905–2913 (2017)
25. Ng, J.Y., Hausknecht, M.J., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: CVPR. pp. 4694–4702 (2015)
26. Poleg, Y., Ephrat, A., Peleg, S., Arora, C.: Compact CNN for indexing egocentric videos. In: WACV. pp. 1–9 (2016)
27. Ryoo, M.S., Rothrock, B., Matthies, L.H.: Pooled motion features for first-person videos. In: CVPR. pp. 896–904 (2015)
28. Singh, S., Arora, C., Jawahar, C.V.: First person action recognition using deep learned descriptors. In: CVPR. pp. 2620–2628 (2016)
29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS. pp. 3104–3112 (2014), http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks
30. Xiao, S., Yan, J., Yang, X., Zha, H., Chu, S.M.: Modeling the intensity function of point process via recurrent neural networks. In: AAAI. pp. 1597–1603 (2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14391
31. Zhang, M., Ma, K.T., Lim, J., Zhao, Q., Feng, J.: Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks. In: CPVR. pp. 3539–3548 (2017)
32. Zhou, Y., Ni, B., Hong, R., Yang, X., Tian, Q.: Cascaded interactional targeting network for egocentric video analysis. In: CVPR. pp. 1904–1913 (2016)
33. Zhu, X., Jia, X., Wong, K.K.: Pixel-level hand detection with shape-aware structured forests. In: ACCV. pp. 64–78 (2014)