# Jointly Recognizing Object Fluents and Tasks in Egocentric Videos

Yang Liu[1*], Ping Wei[2,1*], and Song-Chun Zhu[1]
[1]University of California, Los Angeles, USA
[2]Xi'an Jiaotong University, China
yangliu2014@ucla.edu, pingwei@xjtu.edu.cn, sczhu@stat.ucla.edu

## Abstract

*This paper addresses the problem of jointly recognizing object fluents and tasks in egocentric videos. Fluents are the changeable attributes of objects. Tasks are goal-oriented human activities which interact with objects and aim to change some attributes of the objects. The process of executing a task is a process to change the object fluents over time. We propose a hierarchical model to represent tasks as concurrent and sequential object fluents. In a task, different fluents closely interact with each other both in spatial and temporal domains. Given an egocentric video, a beam search algorithm is applied to jointly recognizing the object fluents in each frame, and the task of the entire video. We collected a large scale egocentric video dataset of tasks and fluents. This dataset contains 14 categories of tasks, 25 object classes, 21 categories of object fluents, 809 video sequences, and approximately 333,000 video frames. The experimental results on this dataset prove the strength of our method.*

## 1. Introduction

Egocentric vision has attracted a growing attention with the advance of wearable camera technologies, such as smart glasses and virtual reality headsets. The wearable cameras mounted on the head enable a user to record the videos from the first-person view while performing daily tasks.

Two significant issues related to egocentric vision are recognizing tasks and recognizing object fluents. A task is a goal-oriented human activity which interacts with the objects in an environment and changes some attributes of the objects, such as *mop floor*, *make coffee*, and *microwave food*. A fluent is a time-varying attribute of an object or a group of objects, and its values are the specific states of the attribute [8, 19], as shown in Fig. 1. For example, a floor's fluent takes the values *dirty* and *clean* over time as the floor is mopped. In our work, fluents are divided into unary fluents and binary fluents. A unary fluent describes the attribute of a single object, such as *dirty* to the floor and *full* to the mug. A binary fluent describes the attribute of two objects as a whole, such as *fastened* to a lid and a coffee can, *contacting* to a blackboard and an eraser, etc.

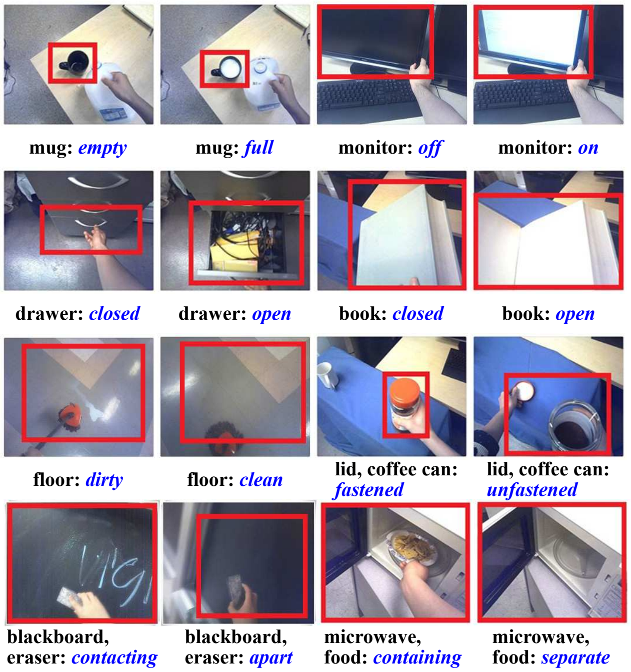
mug: *empty*
mug: *full*
monitor: *off*
monitor: *on*
drawer: *closed*
drawer: *open*
book: *closed*
book: *open*
floor: *dirty*
floor: *clean*
lid, coffee can: *fastened*
lid, coffee can: *unfastened*
blackboard, eraser: *contacting*
blackboard, eraser: *apart*
microwave, food: *containing*
microwave, food: *separate*

Figure 1. Illustration of unary and binary object fluents. The blue italic words describe the object fluents.

One of the most widely-used cues for activity recognition in egocentric videos is the appearance information of related objects [5, 6, 20]. However, in complex goal-oriented tasks, the appearance of the same object often dramatically changes in different phases of the tasks, which may mislead object appearance based activity recognition. For example in Fig. 1, the appearance of the drawer is vastly different before and after the drawer is opened. This phenomenon motivates us to explore new methods to model and recognize tasks.

[*] Yang Liu and Ping Wei are co-first authors.

We propose to model and understand tasks in egocentric videos from a new perspective - the effects that a task causes. A task is a human activity which aims to change some attributes of the objects in an environment. The accomplishment of a task indicates the realization of one or multiple desired fluent changes, which we call the key fluent changes. For example, the task *sweep floor* changes the floor from *dirty* to *clean*, as shown in Fig. 2. With the knowledge that the floor becomes clean from being dirty with some stains, we can reasonably infer that the task *mop floor* might have occurred, even if we did not observe any human activity features. In addition to the fluent of floor, the task *sweep floor* also contains several other key fluents, such as *apart* or *contacting* with respect to *broom* and *trash*, *separate* or *containing* with respect to *dustpan* and *trash*. All these fluents contribute to define and discriminate the task *sweep floor*.

Furthermore, different fluents interact closely with each other both in spatial and temporal domains. In spatial domain, the interaction is presented as fluent concurrence, which means some states of two or more different fluents often occur together. For example, in the task *sweep floor*, a *dirty* floor often means the trash is not contained in the dustpan. The fluent *dirty* with respect to *floor* and the fluent *separate* (not contained) with respect to *dustpan* and *trash* occur together. In temporal domain, the interaction is presented as fluent transition, which means fluents in different tasks transition with different probabilities. For example, in the task *write on blackboard*, it is likely that the blackboard changes from *clean* to *dirty*, but unlikely to change in the opposite direction. This case is just the opposite in the task *clean blackboard*.

In this paper we propose a fluent-based task representation method to jointly recognize object fluents and tasks in egocentric videos. A task is represented as several key object fluents, which interact with each other by means of concurrence in spatial domain and transition in temporal domain. The task, object fluents, video frames, and the relations among them are described with a unified hierarchical graph. Given an egocentric video, a beam search algorithm [36] is adopted to jointly infer the object fluents in each video frame and recognize the task of entire sequence. To evaluate the proposed method, we collected a large scale egocentric video dataset of tasks and fluents in daily activity scenes. The experimental results on this dataset show the effectiveness of our method.

This work makes three major contributions:

1) We represent tasks in egocentric videos from a new perspective - representing tasks with object fluents.
2) We propose a hierarchical model to represent the task, object fluents, and their interaction relations in a unified framework.
3) We collected a new egocentric video dataset of tasks and object fluents. The experiments on this dataset prove the strength of our method.

## 2. Related work.

**Activity modeling and recognition.** Human activity recognition is a classic problem in computer vision and has been intensively studied for decades. Some early studies describe appearance and motion information in 2D images or videos with hand-drafted spatio-temporal features [14, 15, 26, 32, 33]. Recently, deep learned features from neural networks [12, 29, 34] have been applied for activity modeling and produce impressive results. With the advance of motion and depth capture technology, such as Kinect [27], many studies model and analyze human activities in 3D space or RGBD data [13, 36, 37].

To understand the inner contents of human activity videos, some studies model human activities with hierarchical structures [2, 24, 25, 28, 36, 38]. Wei *et al.* [36] proposed a 4D human-object interaction model to jointly recognize human activities and localize objects, and they utilized a dynamic beam search algorithm to solve the inference problem in the hierarchical graph. These hierarchical methods inspire us to represent tasks and fluents in a hierarchical structure.

Different from recognizing the activity of the entire video sequence, some studies recognize activities with partial observation or make early detection [10, 18, 23]. In traditional activity recognition methods, classifiers are trained by encoding the information of the whole video, which is not optimal for activity action recognition with partial observation. Ryoo [23] utilized sequential matching to early recognize human activities with dynamic bag-of-words. Hoai *et al.* [10] used Structured SVM to learn a max-margin early event detector. With the development of deep learning, Recurrent Neural Networks (RNN) such as Long Short Term Memory (LSTM) [9] have been applied to early detection of activities. Ma *et al.* [18] employed LSTM with a designed ranking loss for early activity detection.

**Activity recognition in egocentric video.** Egocentric video analysis has been paid growing attention with the prevalence of egocentric cameras [5, 6, 16, 17, 20, 31]. Activity recognition becomes more challenging in egocentric videos since the movement of camera may lower the performance of the traditional hand-drafted spatial-temporal features such as STIP [14], Dense Trajectory [32], and some deep learned features [29]. Moreover, the human body features become weak or even invisible in egocentric videos. To overcome these difficulties, several semantic egocentric cues have been discovered, such as object cues [5, 6, 20], gaze cues obtained by eye-tracking glasses [6, 16], and hand cues [16, 17, 31]. Li *et al.* [16] elaborately evaluate various mid-level egocentric cues for action recognition and achieved impressive results with different combinations of those cues. Inspired by previous works, descriptors extracted from multi-stream networks [17, 31] encoding different cues have proved efficient.

**Object states and fluents.** Object fluents are used to describe object states [8, 19]. Object state detection and recognition has been recently studied in still images [4, 11, 40]. Isola *et al*. [11] studied the states and transformations of objects /scenes on image collections, and the learned state representations can be extended to different object classes. Fire and Zhu [8] studied the causal relations between human actions and object fluent changes. Fathi and Rehg [7] developed a weakly supervised method to recognize actions and states of manipulated objects before and after the action. Wang *et al*. [35] designed a Siamese network to model precondition states, effect states and their associate actions. Alayrac *et al*. [1] optimized a discriminative cost for joint object state recognition and action localization.

## 3. Task-Fluent Dataset

Since there are no available public datasets for the proposed problem, we collected a new egocentric video dataset about tasks and object fluents. 14 volunteers performed daily tasks in 5 different indoor scenes freely with their own styles. A glasses camera, which can record the videos from the first-person view, is worn by the volunteers when they were performing the tasks. The video frame is at the resolution of 1280 × 960. Some frame samples in the dataset are shown in Fig. 1.

In summary, our dataset consists of 809 videos with approximately 333,000 egocentric video frames. It contains 14 categories of tasks: *sweep floor, mop floor, write on blackboard, clean blackboard, use elevator, pour liquid from jug, make coffee, read book, throw paper, microwave food, use computer, search drawer, move bottle to dispenser*, and *open door*. These tasks involve 25 classes of objects: *broom, dustpan, trash, floor, bucket, mop, chalk, chalk box, blackboard, eraser, elevator, mug, jug, lid, coffee can, book, paper, trash can, microwave, food, monitor, drawer, bottle, dispenser*, and *door*.

In addition to tasks and related objects, this dataset contains 21 categories of object fluents, as shown in Table 1. These fluents are divided into unary fluents and binary fluents. In this dataset, though some objects have the same name of fluent values, they are regarded as different fluent values since their related objects are different. For example, *clean* with respect to *floor* and *clean* with respect to *blackboard* are regarded as different fluent values.

We manually annotated the task label for each video sequence and the object fluent labels in each video frame. A single video frame may contain multiple object fluents and all of the task related fluents are annotated.

Table 1. Object fluent categories.

| **Object** | **Fluent** |
|---|---|
| **Single Object: Unary Fluents** | |
| floor | clean / dirty |
| blackboard | clean / dirty |
| elevator | open / closed |
| microwave | open / closed |
| door | open / closed |
| book | open / closed |
| drawer | open / closed |
| mug | empty / filled / full |
| paper | complete / split |
| monitor | on / off |
| **Two Objects: Binary Fluents** | |
| broom, trash | contacting / apart |
| eraser, blackboard | contacting / apart |
| bottle, dispenser | contacting / apart |
| bucket, mop | containing / separate |
| dustpan, trash | containing / separate |
| chalk box, chalk | containing / separate |
| trash can, paper | containing / separate |
| microwave, food | containing / separate |
| lid, coffee can | fastened / unfastened |
| bottle, dispenser | aligned / misaligned |
| mug, jug | coordinated / uncoordinated |

## 4. Hierarchical Model of Tasks and Fluents

We use a hierarchical graph to represent tasks and object fluents in a unified framework, as shown in Fig. 2. In this representation, a task is composed of several concurrent object fluent changes over time. These object fluents closely interact with each other both in spatial and temporal domains. For example, in Fig. 2, the task *sweep floor* is composed of three categories of fluents. As time flows, the *floor* changes from *dirty* to *clean*; the group of *dustpan* and *trash* changes from *separate* to *containing*, and the group of *broom* and *trash* changes between *apart* and *contacting*. These fluents define the task from the perspective of the effects caused by human activities.

### 4.1. Definition

**Task**. $\mathcal{Y}$ is an alphabet containing $K$ task category labels, such as *sweep floor*, *mop floor*, etc. There are a total of 14 task categories in our work, i.e. $K = 14$. Task recognition is to assign an optimal task label for an input video sequence from the $K$ values in $\mathcal{Y}$.

**Fluent**. Let $\mathcal{F} = \{F_m | m = 1, \dots, M\}$ be the set of all fluent categories, where $M$ is the number of fluent category and 21 in our work, as shown in Table 1. $F_m$ is an alphabet which denotes a fluent category, such as *dirty* or *clean* with respect to *floor*. The elements in $F_m$ are the possible fluent values, such as '*dirty*' and '*clean*'.

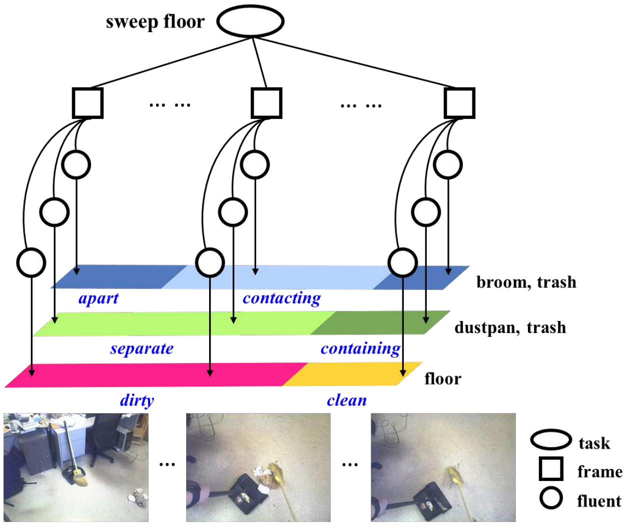

Figure 2. Joint model of tasks and object fluents.

### 4.2. Formulation

Let $X = \{\mathbf{x}_t | t = 1, \dots, T\}$ be a video sequence containing $T$ frames, where $\mathbf{x}_t$ is the video frame at time $t$. Suppose $Y \in \mathcal{Y}$ is the task category label of the sequence $X$. $Z = \{\mathbf{z}_t | t = 1, \dots, T\}$ is the sequence of the fluent labels for the video sequence $X$.

For a specific task class $Y \in \mathcal{Y}$, it has $N_Y$ categories of key fluents $\{F_{Y,n} | n = 1, \dots, N_Y\}$, where $F_{Y,n} \in \mathcal{F}$ is the $n$th key fluent category of the task $Y$. For example, the task *sweep floor* has three categories of key fluents, as the three colorful bars show in Fig.2.

$\mathbf{z}_t = \{z_t^n | n = 1, \dots, N_Y\}$ is the fluent label set of the video frame $\mathbf{x}_t$. The $N_Y$ elements of $\mathbf{z}_t$ correspond to the $N_Y$ key fluents, respectively. $z_t^n \in F_{Y,n}$ is the value of the $n$th key fluent, such as $z_t^n =$ '*clean*' in the fluent with respect to *floor*, $z_t^n =$ '*apart*' in the fluent with respect to *broom* and *trash*, etc.

The score that the video sequence $X$ is interpreted by the task category label $Y$ and the fluent label sequence $Z$ is defined as

$$\begin{aligned} S(X,Y,Z) = & \underbrace{\sum_{t=1}^{T} \sum_{n=1}^{N_Y} \varphi(\mathbf{x}_t, z_t^n, Y) +}_{\text{feature matching}} \\ & \underbrace{\sum_{t=1}^{T} \sum_{n \neq n'}^{N_Y} \phi(Y, z_t^n, z_t^{n'})}_{\text{spatial concurrence}} + \underbrace{\sum_{t=2}^{T} \sum_{n=1}^{N_Y} \psi(Y, z_{t-1}^n, z_t^n)}_{\text{temporal transition}}, \end{aligned} \tag{1}$$

where $\varphi(\cdot)$, $\phi(\cdot)$, and $\psi(\cdot)$ are the feature matching, spatial concurrence, and temporal transition functions, respectively. We elaborate on them as follows.

**Feature matching.** $\varphi(\mathbf{x}_t, z_t^n, Y)$ measures the compatibility between the fluent label $z_t^n$ and the frame feature $\mathbf{x}_t$. Suppose $\mathbf{o}_{Y,n} = \{o_{Y,n}^i | i = 1, ..., |\mathbf{o}_{Y,n}|\}$ is the class label set of the related objects to the $n$th key fluent in the task $Y$, where $|\mathbf{o}_{Y,n}|$ is the related object class number. $|\mathbf{o}_{Y,n}|$ is 1 or 2 in our dataset. $\hat{\mathbf{o}}_{Y,n}$ is a set of bounding boxes of the objects in $\mathbf{o}_{Y,n}$.

$\varphi(\mathbf{x}_t, z_t^n, Y)$ is rewritten as

$$\varphi(\mathbf{x}_t, z_t^n, Y) = \varphi_1(\mathbf{x}_t, \mathbf{o}_{Y,n}) + \varphi_2(\mathbf{x}_t, z_t^n, \hat{\mathbf{o}}_{Y,n}). \tag{2}$$

$\varphi_1(\mathbf{x}_t, \mathbf{o}_{Y,n})$ is the object detection term, which describes the occurrence belief of the fluent-related objects in the video frame at time $t$. We trained object detectors by fine-tuning Faster R-CNN [21] on our dataset and use the trained detectors to generate the object detection probabilities. Suppose $p(o_{Y,n}^i | \mathbf{x}_t)$ is the detection probability of the object class $o_{Y,n}^i$. The object detection term is

$$\varphi_1(\mathbf{x}_t, \mathbf{o}_{Y,n}) = \frac{1}{|\mathbf{o}_{Y,n}|} \sum_{i=1}^{|\mathbf{o}_{Y,n}|} \ln p(o_{Y,n}^i | \mathbf{x}_t). \tag{3}$$

$\varphi_2(\mathbf{x}_t, z_t^n, \hat{\mathbf{o}}_{Y,n})$ is the fluent labeling term, which measures the compatibility between the object state feature and the fluent label. We define the fluent area as a bounding box which covers all the bounding boxes in $\hat{\mathbf{o}}_{Y,n}$ with the minimum size. Using the features in the fluent areas, we train a classifier for each fluent category with VGG-16 model [30]. Suppose $p(z_t^n | \mathbf{x}_t, \hat{\mathbf{o}}_{Y,n})$ is the classification probability output by the fluent classifier. The fluent labeling term is defined as

$$\varphi_2(\mathbf{x}_t, z_t^n, \hat{\mathbf{o}}_{Y,n}) = \ln p(z_t^n | \mathbf{x}_t, \hat{\mathbf{o}}_{Y,n}). \tag{4}$$

**Spatial concurrence.** $\phi(Y, z_t^n, z_t^{n'})$ measures the compatibility between different fluent categories $z_t^n$ and $z_t^{n'}$ in task Y. For each task category, we compute the average prior frequencies that the values of different fluents occur together from the training videos. Suppose $q(z_t^n, z_t^{n'})$ is the average prior frequency that $z_t^n$ and $z_t^{n'}$ occur together. The spatial concurrence term is defined as:

$$\phi(Y, z_t^n, z_t^{n'}) = \ln q(z_t^n, z_t^{n'}) \tag{5}$$

**Temporal transition.** $\psi(Y, z_{t-1}^n, z_t^n)$ measures the continuity and transition relations of the fluent values $z_{t-1}^n$ and $z_t^n$ in two adjacent frames. We use a Markov chain to model the fluent value transitions. Suppose $r(z_{t-1}^n, z_t^n)$ is the probability of the transition from $z_{t-1}^n$ to $z_t^n$. The temporal transition term is

$$\phi(Y, z_{t-1}^n, z_t^n) = \ln r(z_{t-1}^n, z_t^n). \tag{6}$$

The transition probabilities are learned from video samples of each task.

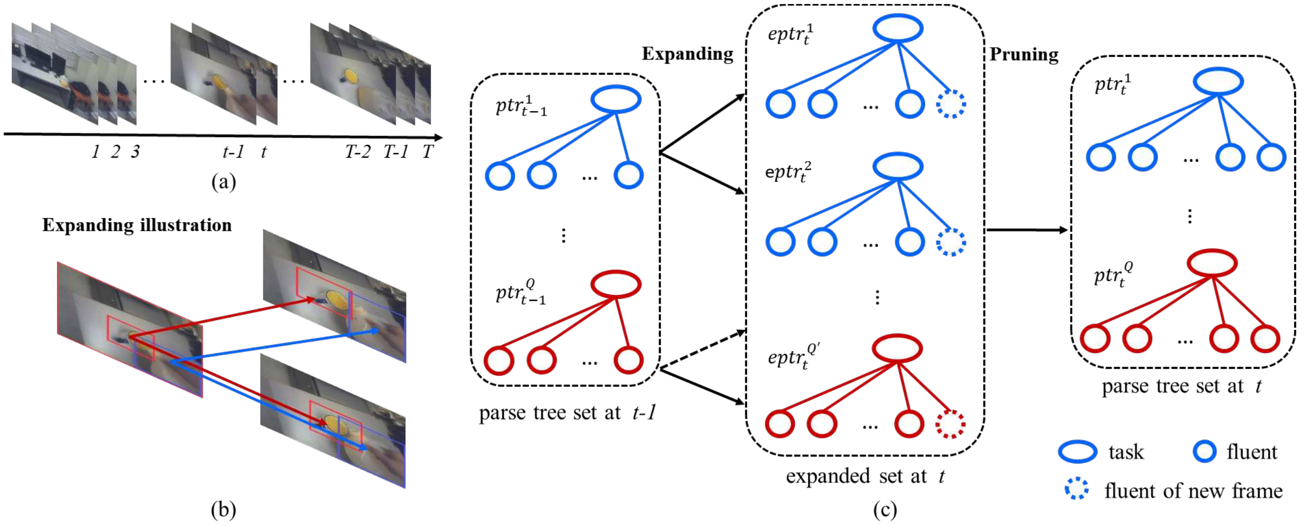

Figure 3. Dynamic programming beam search for jointly recognizing fluents and tasks. (a)The input video sequence. (b) Illustration of expanding one parse tree to new parse trees. (c) Parse tree expanding and pruning in one iteration.

# 5. Inference

Given a video sequence $X = \{\mathbf{x}_t | t = 1, \ldots, T\}$ , the goal is to compute its task label $Y^*$ and the fluent label sequence $Z^* = \{\mathbf{z}_t^* | t = 1, \ldots, T\}$ that maximize the score function $S(X, Y, Z)$, which is formulated as

$$(Y^*, Z^*) = \underset{Y,Z}{\operatorname{argmax}}\ S(X, Y, Z) \tag{7}$$

The solution to Eq. 7 is a parse tree, in which the root node is the task label, and the leaf nodes are the sequence of fluent labels. To obtain the optimal result, an intuitive idea is to examine all the possible parse trees and output the one with the maximum score as the optimal result. However, the huge size of the parse tree space makes such exhaustive search method inapplicable. Inspired by the work in [36], we adopt a beam search algorithm to solve the above problem 7.

The general procedures of this algorithm include: (1) proposing multiple object bounding boxes with pre-trained object detectors in each video frame; (2) generating possible interpretations of the current frame by expanding the parse trees of previous frames; (3) pruning the parse trees with smaller scores and keeping the rest as the possible interpretations to the current video sequences. The algorithm is illustrated in Fig. 3.

At the first frame, we enumerate all possible task labels and the corresponding fluent labels based on the task-related object proposals to initialize a parse tree set. At time $t-1$, suppose $PTR_{t-1} = \{ptr_{t-1}^i | i = 1, \ldots, Q\}$ is the parse tree set with $Q$ parse trees of labeling the video clip from time 1 to time $t-1$. With the frame at time $t$, we expand every parse tree $ptr_{t-1}^i \in PTR_{t-1}$ by adding new task-related object proposals and key fluent values in the new frame, as shown in Fig. 3 (b). After expanding all $ptr_{t-1}^i \in PTR_{t-1}$, we obtain an expanded parse tree set $\{eptr_t^1, ..., eptr_t^{Q'}\}$ for the video clip from time 1 to time $t$, as shown in Fig. 3 (c).

The expanded set $\{eptr_t^1, ..., eptr_t^{Q'}\}$ often contains large number of parse trees and many of them with small scores are misleading interpretations to the video clip. To increase the computational efficiency and improve the performance, we sort all the parse trees in the expanded set by their scores and keep the first $Q$ trees with the largest scores. In this way, we obtain the parse tree set $PTR_t = \{ptr_t^i | i = 1, \ldots, Q\}$ at time $t$, as is shown in Fig. 3 (c). This expanding and pruning process are iterated to the last frame of the sequence. The optimal $(Y^*, Z^*)$ for the entire sequence is the parse tree in $PTR_T$ with the maximal score.

# 6. Experiments

## 6.1. Experiments Setup

We test our method on our newly collected Task-Fluent Dataset and the evaluations include fluent recognition and task recognition. We use recognition accuracy as the evaluation metric. For fluent recognition, the accuracy is defined as the ratio of the correctly recognized fluent state number to the total testing fluent state number in all testing video frames. For the task recognition, the accuracy is defined as the ratio of correctly recognized video number to the total testing video number. The ratios for the training, validation, and testing video numbers are 0.5, 0.25, and 0.25, respectively.

Table 2. Comparison of different methods for unary fluents.

| Method | **floor** | **board** | **elevator** | **mcwave** | **door** | **book** | **drawer** | **mug** | **paper** | **monitor** |
|---|---|---|---|---|---|---|---|---|---|---|
| | *clean* | *clean* | *open* | *open* | *open* | *open* | *open* | *empty* | *complete* | *on* |
| | *dirty* | *dirty* | *closed* | *closed* | *closed* | *closed* | *closed* | *filled/full* | *split* | *off* |
| SFC | 0.79 | 0.66 | 0.98 | 0.79 | 0.78 | 0.93 | 0.75 | 0.85 | 0.91 | 0.97 |
| Spatial LSTM | 0.85 | 0.67 | 0.99 | 0.90 | 0.83 | 0.90 | 0.82 | 0.87 | 0.86 | **0.99** |
| Two-s LSTM | 0.84 | 0.70 | 0.99 | 0.91 | **0.94** | 0.88 | 0.67 | **0.89** | 0.90 | **0.99** |
| Our Method | **0.90** | **0.76** | **1.00** | **0.92** | 0.86 | **0.99** | **0.99** | 0.66 | **0.94** | 0.98 |

Table 3. Comparison of different methods for binary fluents.

| Method | **broom** | **eraser** | **bucket** | **dustpan** | **box** | **mug** | **trashcan** | **mcwave** | **lid** | **bottle** | **bottle** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **trash** | **board** | **mop** | **trash** | **chalk** | **jug** | **paper** | **food** | **coffcan** | **dispens** | **dispens** |
| | *contact* | *contact* | *contain* | *contain* | *contain* | *coord* | *contain* | *contain* | *fastened* | *contact* | *aligned* |
| | *apart* | *apart* | *sep* | *sep* | *sep* | *uncoord* | *sep* | *sep* | *unfstn* | *apart* | *misalign* |
| SFC | 0.58 | 0.89 | 0.87 | 0.67 | **0.84** | 0.86 | 0.93 | 0.89 | 0.82 | 0.80 | 0.87 |
| Spatial LSTM | 0.51 | 0.92 | 0.77 | 0.86 | 0.74 | 0.80 | 0.94 | **0.94** | 0.78 | 0.82 | **0.92** |
| Two-s LSTM | 0.73 | 0.89 | 0.86 | 0.81 | 0.78 | 0.71 | **0.99** | 0.89 | **0.87** | 0.83 | 0.88 |
| Our Method | **0.88** | **0.95** | **0.98** | **0.90** | 0.81 | **0.95** | 0.93 | **0.94** | 0.58 | **0.89** | **0.92** |

* board = blackboard; box = chalk box; mcwave = microwave; coffcan = coffee can; coord = coordinate; sep = separate; unfstn = unfastened.

Table 4. Comparison of different methods for fluent recognition.

| Method | Unary | Binary | Overall |
|---|---|---|---|
| SFC | 0.838 | 0.836 | 0.837 |
| Spatial LSTM | 0.859 | 0.820 | 0.843 |
| Two-s LSTM | 0.861 | 0.846 | 0.855 |
| Our Method | **0.909** | **0.869** | **0.896** |

For object detection and proposal in Eq. 2, we finetune Faster R-CNN [21] on our dataset using VGG-16 [30] trained on ILSVRC2012 [22] as pre-trained model. The number of iterations we set for 2 stage training process are 80K, 40K, 80K, 40K. The confidence threshold is 0.5 and the non-maximum suppression threshold is 0.6. Some object proposal results are shown in Fig. 4.

We use the ground truth object bounding boxes and fluent labels from the dataset to train the fluent feature matching term in Eq. 2. For each fluent category, we crop the fluent areas from video frames and trained classifiers by finetuning the VGG-16 model [30].

## 6.2. Fluent Recognition

We compare our method with several baseline methods. (1) Single frame classification (SFC). This method takes fluent recognition as a multi-class image classification problem. We train an classifier by fine-tuning VGG-16 model [30] and replace *soft-max* with *sigmoid* to adapt for multiclass output. We select the threshold = 0.1 by maximizing the fluent classification accuracy on validation set. (2) Spatial stream LSTM (Spatial LSTM). This method uses a LSTM network [9] with image appearance features for fluent recognition. Based on the network we train in method SFC, for each frame in the video, we retrieve 4096 dimensional features from *fc2* as the appearance descriptors. The videos are trimmed into video clips with length L = 60. The LSTM network was built by stacking three bidirectional LSTM layers. (3) Two-stream LSTM (Two-s LSTM). Inspired by Two-stream networks [29] and two recently studies which applied multi-stream to egocentric video action recognition [17, 31], we use the two-stream LSTM for fluent recognition. We train a multi-class classification network on optical flow features with a network architecture similar to VGG-CNN-M [3] as the temporal stream LSTM. We build a two-stream LSTM network by combining the temporal LSTM and the spatial stream LSTM.

Table 2 and Table 3 show the accuracy comparison on each fluent category. Our method achieves best performance on most fluent categories, which demonstrates the advantage of our model.

Table 4 shows the overall accuracy comparison of our method with other baseline methods. It also separately shows the unary fluent recognition accuracy and binary fluent accuracy. Our method outperforms other baseline methods in each item, which demonstrates the advantage and effectiveness of our joint modeling method. This table also shows that, to each method, the recognition accuracy on unary fluents is higher than that of binary fluents. One explanation is that the binary fluents are related to the spatial relationships between two objects. It is difficult to encode such spatial relationships on 2D images.

Some qualitative results are shown in Fig. 4. In most of the cases, our method can identify the task category, locate the objects, and recognize the fluents correctly. It should be noted that our method can infer the object fluents without the fluent features. For example, in the first image of the task *sweep floor*, although *trash* is not in the frame and *floor* is not detected, our method can still infer the correct fluent label by reasoning about the spatial concurrence and tempo-

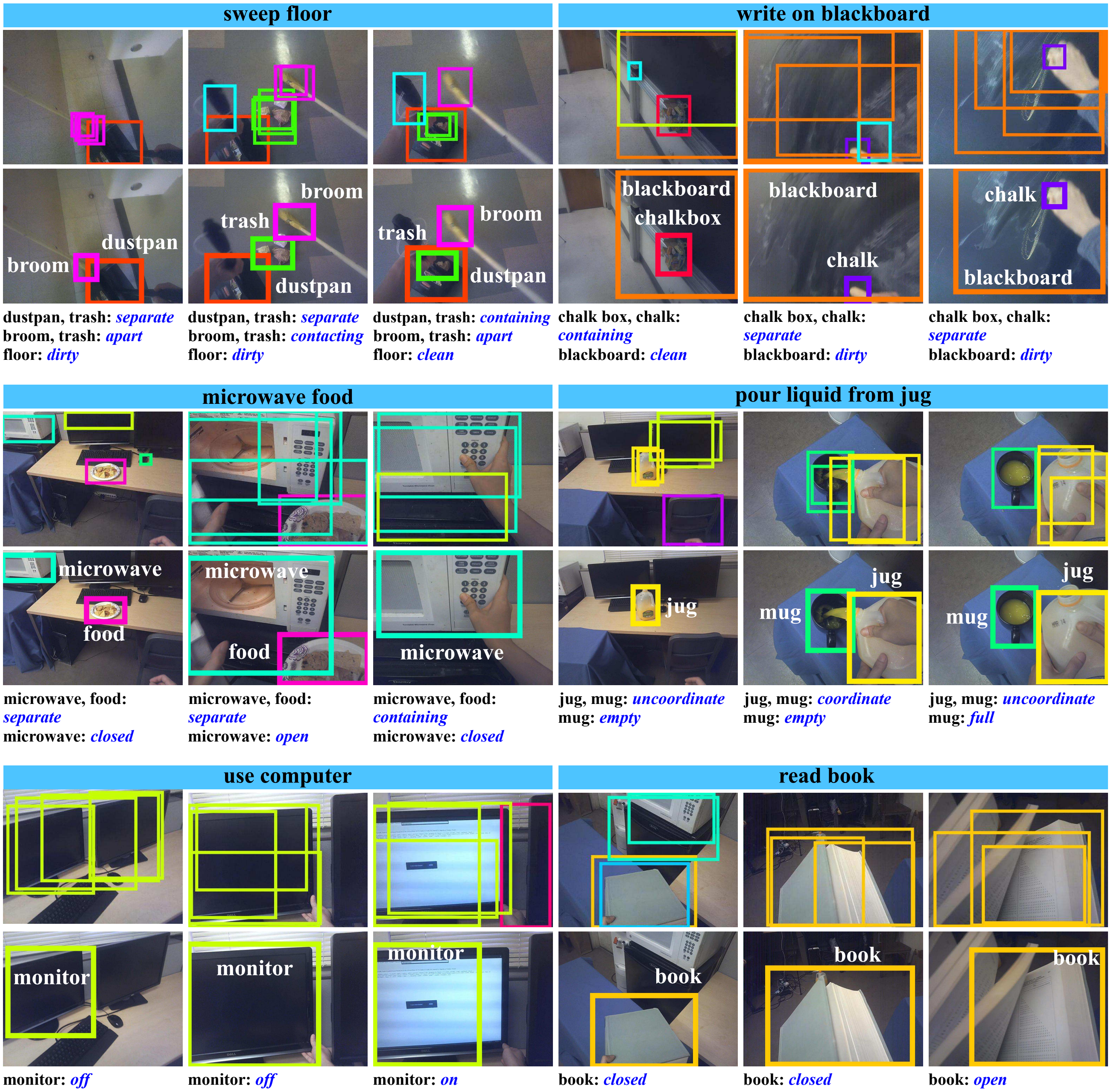

Figure 4. Some results of joint recognition of object fluents and tasks. The labels in the bars are the tasks. For every two rows, the first row shows the object proposals; the second row shows our output results, and the descriptions below the images are about the fluent recognition.

ral transition relations among fluents. In the task *read book*, task-unrelated object *microwave* is detected in the frame. Our joint model correctly recognizes the task and outputs correct fluent label of *book*. This illustrates the advantage of the joint modeling framework of tasks and fluents.

### 6.3. Task Recognition

For task recognition, we compare our method with two baseline methods. (1) CNN hit@1. This method uses the single frame features with hit@1 rule [12] to recognize tasks of videos. As described in Karpathy's work on video classification [12], we train a single frame classifier by fine-tuning VGG-16 model with a 14-dimensional output layer at top. In testing, each frame can output one task label, we average those predictions of frames and output one explicit task label for each sequence. (2) LSTM. As the approach suggested in [39], we retrieve the output of the second fully connected layer of single frame model we train in CNN

Table 5. Comparison of different methods for tasks recognition

| Method | Average |
|---|---|
| CNN hit@1 | 0.90 |
| LSTM | 0.87 |
| Our Method | **0.96** |

hit@1. The video-level three stacked LSTM networks are trained with video clips of length 10 for task recognition.

As shown in Table 5, our method outperforms the baseline methods, which shows that the fluent-based representation of complex tasks is reasonable and effective. Table 5 shows that the CNN hit@1 method achieves a better accuracy than the LSTM method incorporating the motion information. The main reason is that the egocentric videos contain massive motion features which are not related to tasks but caused by the irregular movement of the user's head. Such motion information will mislead task recognition.

Since our model has a hierarchical structure and our inference algorithm is an online framework, our method can recognize the task with partial observation of a video sequence, i.e. early recognition of tasks. We compare our method with above 2 baseline methods at different observation ratios of each video sequence. At each ratio point, each method is fed with a video clip from the first frame to the frame at the position corresponding to the ratio length of the whole video length.

Fig. 5 shows the accuracy comparison at different sequence length ratios. Our method outperforms the other baseline methods at every observation ratio. This figure shows that with fewer observation video frames our method can achieve a comparable accuracy with other methods at a lager observation ratio. This is mainly because our fluent-based method uses features of objects related to the tasks rather than the entire image features.

## 7. Conclusion

In this paper, we study a new problem of jointly recognizing object fluents and tasks in egocentric videos. We propose a unified fluent-based task representation framework, in which tasks are modeled with object fluents. In each task, different fluents closely interact with each other by means of spatial concurrence and temporal transition. Given a testing egocentric video, a beam search algorithm is used to jointly recognize the object fluents in each frame, and the task of the entire video. We collect a large-scale egocentric video dataset including various fluents and tasks with detailed annotations. Our experiments have shown that our model outperforms the baseline methods which proves the strength of our model. Our future work will focus on the continuous fluents, object independent fluents, and the fluent-based tasks in robotics.

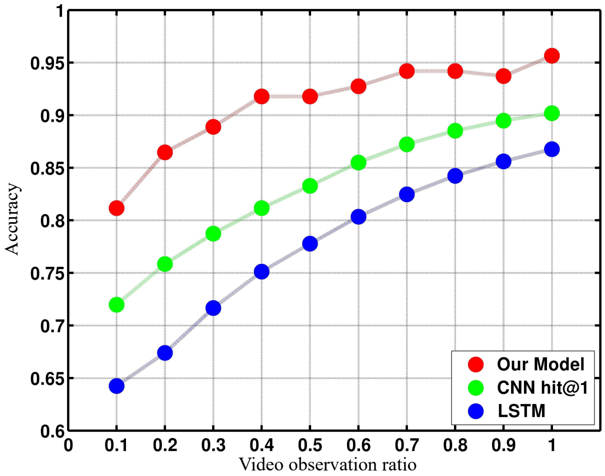

Figure 5. Comparison of our method with baselines.

## Acknowledgment

This research was supported by grants DARPA XAI project N66001-17-2-4029, ONR MURI project N00014-16-1-2007, NSF IIS-1423305, and NSFC 61503297.

## References

[1] J.-B. Alayrac, J. Sivic, I. Laptev, and S. Lacoste-Julien. Joint discovery of object states and manipulation actions. In *International Conference on Computer Vision*, 2017.

[2] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *IEEE International Conference on Computer Vision*, 2011.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[4] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[5] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *International Conference on Computer Vision*, 2011.

[6] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision*, 2012.

[7] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[8] A. Fire and S.-C. Zhu. Learning perceptual causality from video. *ACM Transactions on Intelligent Systems and Technology*, 7(2):23:1–23:22, 2015.

[9] A. Graves. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 5–13. Springer, 2012.

[10] M. Hoai and F. De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014.

[11] P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[13] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *International Journal of Robotics Research*, 32(8):951–970, 2013.

[14] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[16] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[17] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[18] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[19] E. T. Mueller. *Commonsense Reasoning: An Event Calculus Based Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2015.

[20] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[23] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE International Conference on Computer Vision*, 2011.

[24] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *IEEE International Conference on Computer Vision*, 2009.

[25] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *IEEE Workshop on Motion and Video Computing*, 2008.

[26] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *International Conference on Pattern Recognition*, volume 3, 2004.

[27] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[28] Z. Si, M. Pei, B. Yao, and S.-C. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *IEEE International Conference on Computer Vision*, 2011.

[29] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] S. Singh, C. Arora, and C. Jawahar. First person action recognition using deep learned descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[32] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[33] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, 2013.

[34] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[35] X. Wang, A. Farhadi, and A. Gupta. Actions ˜ transformations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[36] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. Modeling 4d human-object interactions for joint event segmentation, recognition, and object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1165–1179, 2017.

[37] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu. Concurrent action detection with structural prediction. In *IEEE International Conference on Computer Vision*, 2013.

[38] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu. Multi-view people tracking via hierarchical trajectory composition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[39] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[40] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision*, 2014.