Anticipating Human Activities using Object Affordances for Reactive Robotic Response

Hema S. Koppula and Ashutosh Saxena, *Member, IEEE*

Abstract—An important aspect of human perception is anticipation, which we use extensively in our day-to-day activities when interacting with other humans as well as with our surroundings. Anticipating which activities will a human do next (and how) can enable an assistive robot to plan ahead for reactive responses. Furthermore, anticipation can even improve the detection accuracy of past activities. The challenge, however, is two-fold: We need to capture the rich context for modeling the activities and object affordances, and we need to anticipate the distribution over a large space of future human activities. In this work, we represent each possible future using an anticipatory temporal conditional random field (ATCRF) that models the rich spatial-temporal relations through object affordances. We then consider each ATCRF as a particle and represent the distribution over the potential futures using a set of particles. In extensive evaluation on CAD-120 human activity RGB-D dataset, we first show that anticipation improves the state-of-the-art detection results. We then show that for new subjects (not seen in the training set), we obtain an activity anticipation accuracy (defined as whether one of top three predictions actually happened) of 84.1%, 74.4% and 62.2% for an anticipation time of 1, 3 and 10 seconds respectively. Finally, we also show a robot using our algorithm for performing a few reactive responses.[†]

Index Terms—RGBD Data, 3D Activity Understanding, Human Activity Anticipation, Machine Learning, Robotics Perception.

1 Introduction

For many applications it is important to be able to detect what a human is currently doing as well as *anticipate* what she is going to do next and how. The former ability is useful for applications such as monitoring and surveillance, but we need the latter for applications that require reactive responses, for example, an assistive robot (see Figure 1). In this paper, our goal is to use anticipation for predicting future activities as well as improving detection (of past activities).

There has been a significant amount of work in detecting human activities from 2D RGB videos [1], [2], [3], from inertial/location sensors [4], and more recently from RGB-D videos [5], [6], [7]. The primary approach in these works is to first convert the input sensor stream into a spatio-temporal representation, and then to infer labels over the inputs. These works use different types of information, such as human pose, interaction with objects, object shape and appearance features. However, these methods can be used only to predict the labeling of an observed activity and cannot be used to anticipate what can happen next and how.

Our goal is to predict the future activities as well as the details of how a human is going to perform them in short-term (e.g., 1-10 seconds). For example, if a robot has seen a person move his hand to a coffee mug, it is possible he would move the coffee mug to a few potential places such as his mouth, to a kitchen sink or just move it to a different location on the table. If a robot can anticipate this, then it would rather not start pouring milk into the coffee when the person is moving his hand towards the mug, thus avoiding a spill. Such scenarios happen in several other settings, for example, manufacturing scenarios in future co-robotic settings (e.g., [8], [9]).

Activities often have a hierarchical structure where an activity is composed of a sequence of sub-activities and involve interactions with certain objects. For example, a cup is used in the *drinking* activity which is composed of a sequence of *reach*, *move* and *drink* sub-activities. Therefore, we can anticipate the future by observing the sub-activities performed in the past and reasoning about the future based on the structure of activities and the functionality of objects being used (also referred to as object affordances [10]). For example, in Figure 1, on seeing a person carrying a bowl and walking towards the refrigerator, one of the most likely future actions are to reach the refrigerator, open it and place the bowl inside.

For anticipating the future, we need to predict how the future sub-activities will be performed in terms of motion trajectories of the objects and humans. In order to do this, we propose the use of object affordances and model them in terms of the relative position of the object with respect to the human and the environment[1] and their temporal motions trajectories during the activity, as described in Section 4. Modeling trajectories not only helps in discriminating the activities,[2] but is also useful for the robot to reactively plan motions in the workspace.

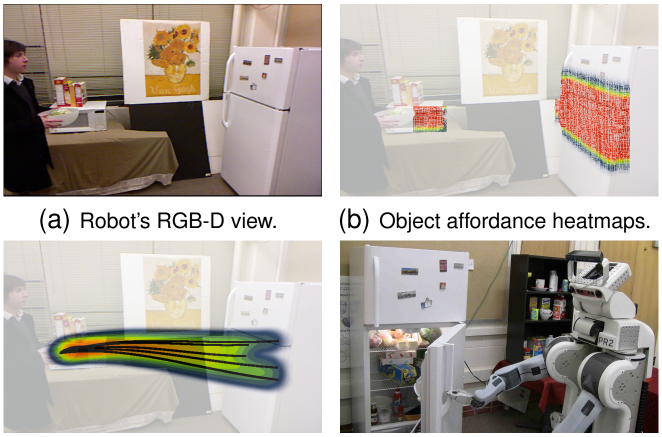
In our work, we use a conditional random field based on [5] (see Figure 2) to model the spatio-temporal structure of activities, as described in Section 5.1. For anticipation, we present an anticipatory temporal conditional random field (ATCRF), where we model the past with the CRF described above but augmented with the trajectories and with nodes/edges representing the object affordances, sub-activities, and trajectories in the future. Since there are many possible futures, each

• *H. S. Koppula and A. Saxena are with the Computer Science Department at Cornell University, NY 14853. E-mail: {hema,asaxena}@cs.cornell.edu*

[†] *Parts of this work have been published at ICML 2013 and RSS 2013. The paper at RSS 2013 received the best student paper award and was the runner-up for the best paper award.*

[1] For example, a *drinkable* object is found near the mouth of the person performing the *drinking* activity and a *placeable* object is near a stable surface in the environment where it is being placed.

[2] For example, in stirring activity, the target position of the stirrer is immaterial but the circular trajectory motion is.


(a) Robot's RGB-D view. (b) Object affordance heatmaps.

(c) Trajectories heatmap. (d) Robot opening the door.

Fig. 1: **Reactive robot response through anticipation:** Robot observes a person holding an object and walking towards a fridge (a). It uses our ATCRF to anticipate the affordances (b), and trajectories (c). It then performs an anticipatory action of opening the door (d).

ATCRF represents only one of them. In order to find the most likely ones, we consider each ATCRF as a particle and propagate them over time, using the set of particles to represent the distribution over the future possible activities. One challenge is to use the discriminative power of the CRFs (where the observations are continuous and labels are discrete) for also producing the generative anticipation—labels over sub-activities, affordances, and spatial trajectories.

We evaluate our anticipation approach extensively on CAD-120 human activity dataset [5], which contains 120 RGB-D videos of daily human activities, such as *having meal*, *microwaving food*, *taking medicine*, etc. We first show that anticipation improves the detection of *past* activities: 85.0% with vs 82.3% without. Our algorithm obtains an activity anticipation accuracy (defined as whether one of top three predictions actually happened) of (84.1%,74.4%,62.2%) for predicting (1,3,10) seconds into the future. Videos showing our experiments and *code* are available at: `http://pr.cs.cornell.edu/anticipation`

The main contributions of this paper are:

- While most previous works consider activity detection, we consider *anticipation*.
- We consider rich contextual relations based on object affordances in RGB-D videos.
- We propose ATCRFs, where each particle represents a CRF.
- We consider joint temporal segmentation and labeling using our approach.
- We test our method on a dataset containing long-term activities, and also test in robotic experiments.

The rest of the paper is organized as follows. We start with a review of the related work in Section 2 and provide an overview of our methodology in Section 3. We describe the affordances in Section 4 and model in Section 5. Section 6 describes the features and Section 7 describes the learning and inference algorithms. We present the experimental results in Section 8 and finally conclude the paper in Section 9.

2 Related Work

Activity Detection. In recent years, much effort has been made to detect human activities from still images as well as videos (e.g., [11], [12]). Many methods have been proposed to model the temporal structure of low-level features extracted from video, e.g., histograms of spatiotemporal filter responses. This includes both discriminative [1], [13], [14], [15] and generative models [16], [17]. Another approach is to represent activities as collections of semantic attributes [18], [19], [20], [21]. These methods use an intermediate level of representation such as the presence or absence of semantic concepts (e.g., scene types, actions, objects, etc.) in order to generalize to unseen instances. There are also a few recent works which address the task of early recognition [22], [23]. We refer the reader to [24] for a comprehensive survey of the field and discuss works that are closely related to ours.

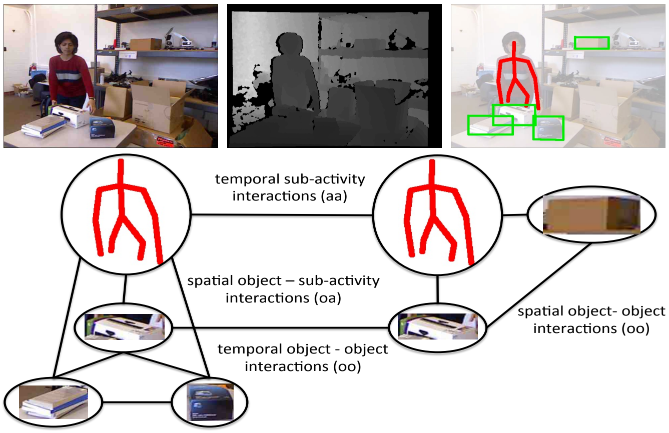

temporal sub-activity interactions (aa)
spatial object – sub-activity interactions (oa)
spatial object- object interactions (oo)
temporal object - object interactions (oo)

Fig. 2: An example activity from the CAD-120 dataset (top row) and one possible graph structure (bottom row). Top row shows the RGB image (left), depths (middle), and the extracted skeleton and object information (right). (Graph in the bottom row shows the nodes at only the temporal segment level, the frame level nodes are not shown.)

Some works use human pose for action recognition by detecting local pose features [25] and modeling the spatial configuration between human body parts and objects [5], [26], [27], [28], [29]. Recent availability of inexpensive RGB-D sensors has enabled significant improvement in scene modeling [30], [31] and estimation of human poses [32], [33]. This, together with depth information, has enabled some recent works [6], [7], [34] to obtain good action recognition performance. However, these methods only address detection over small time periods, where temporal segmentation (i.e., knowledge of the spatio-temporal structure) is not an issue.

Koppula, Gupta and Saxena (KGS, [5]) proposed a model to jointly predict sub-activities and object affordances by taking into account spatio-temporal interactions between human poses and objects over longer time periods. However, KGS found that not knowing the graph structure (i.e., the correct temporal segmentation) decreased the performance significantly. This is because the boundary between two sub-activities is often not very clear, as people often start performing the next sub-activity before finishing the current sub-activity. Moreover, all these work only detect the activities and affordance after the action is performed. None of these methods can *anticipate* what is going to happen next.

Temporal Segmentation. In activity detection from 2D videos, much previous work has focussed on short video clips, assuming that temporal segmentation has been done apriori. It has been observed that temporal boundaries of actions are not precisely defined in practice, whether they are obtained automatically using weak-supervision [35] or by hand [36]. These works represent the action clips by an orderless bag-of-features and try to improve classification of the action clips by refining their temporal boundaries. However, they only model the temporal extent of actions, not their temporal structure.

Some recent effort in recognizing actions from longer video sequences take an event detection approach [15], [37], [38], [39], where they evaluate a classifier function at many different segments of the video and then predict event presence. Similarly, change point detection methods [40], [41] perform a sequence of change-point analysis in a sliding window along the time dimension. However, these methods only detect *local* boundaries and tend to over-segment complex actions which often contain many changes in local motion statistics.

Some previous works consider joint segmentation and recognition by defining dynamical models based on kinematics [42], [43], but these works do not model the complex human-object interactions. [44] and [45] perform activity classification and clustering respectively but do not consider temporal context. In contrast, our application requires modeling of the temporal context (as compared to just spatial).

In contemporary work, [46] performs joint segmentation and labeling using a structural model which takes into account the activity durations, motion features and context features. They use a nonlinear dynamical model to obtain action segments which are then merged into activities of interest. Wang et al. [47] predict sports moves and human activity in TV episodes by solving a bilinear program to jointly estimate the structure of an MRF graph and perform MAP inference. However, these works do not anticipate the future activities.

Anticipation of Human Actions. Anticipation or forecasting future human actions has been the focus of few recent works. Maximum entropy inverse reinforcement learning was used by [48], [49], [50] to obtain a distribution over possible human navigation trajectories from visual data, and also used to model the forthcoming interactions with pedestrians for mobile robots [48], [50]. However, these works focus only on human actions which are limited to navigation trajectories. Wang et al. [51] propose a latent variable model for inferring unknown human intentions, such as the target ball position in a robot table tennis scenario, and Dragan et al. [52] use inverse reinforcement learning to predict future goal for grasping an object. In comparison, we address the problem of anticipation of human actions at a fine-grained level of how a human interacts with objects in more involved activities such as *microwaving food* or *taking medicine* compared to the generic navigation activities or task-specific trajectories.

Learning Algorithms. Our work uses probabilistic graphical models to capture rich context. Such frameworks as HMMs [53], [54], DBNs [55], CRFs [5], [56], [57], semi-CRFs [58] have been previously used to model the temporal structure of videos and text. While these previous works maintain their template graph structure over time, in our work, new graph structures are possible. Works on semi-Markov models [58], [59] are quite related as they address the problem of finding the segmentation along with labeling. However, these methods are limited since they are only efficient for feature maps that are additive in nature. We build upon these ideas where we use additive feature maps only as a close approximation to the graph structure and then explore the space of likely graph structure by designing moves. We show that this improves performance while being computationally efficient.

For anticipation, we use importance sampling for efficient estimation of the likelihood of the potential future activities. Particle filters have been applied with great success to a variety of state estimation problems including object tracking [60], [61], mobile robot localization [62], [63], people tracking [64], etc. However, the worst-case complexity of these methods grows exponentially in the dimensions of the state space, it is not clear how particle filters can be applied to arbitrary, high-dimensional estimation problems. Some approaches use factorizations of the state space and apply different representations for the individual parts of the state space model. For example, Rao-Blackwellised particle filters sample only the discrete and non-linear parts of a state estimation problem. The remaining parts of the states are solved analytically conditioned on the particles by using Kalman filters [64], [65], [66], [67]. In our work, each of our particles is a CRF that models rich structure and lies in a high-dimensional space.

3 Overview

Our goal is to anticipate what a human will do next given the current observation of his pose and the surrounding environment. These observations are from RGB-D videos recorded with a Kinect sensor. From these videos, we obtain the human pose using the Openni's skeleton tracker [68] and extract the tracked object point clouds using SIFT feature matching as described in KGS [5].[3] Note that we infer the object affordances based on its usage in the activity and do not require the object category labels. We discuss the effect of knowing the object categories on the anticipation performance in Section 5.3.

Since activities happen over a long time horizon, with each activity being composed of sub-activities involving different number of objects, we perform segmentation in time, as described in Section 7.1, such that each temporal segment represents one sub-activity. We model the activity using a spatio-temporal graph (a CRF), as shown in Figure 3-left. The extracted human pose and objects form the nodes in this graph, and the edges between them represent their interactions, as described in Section 5.1. However, this graph can only model the present observations. In order to predict the future, we augment the graph with an 'anticipated' temporal segment, with anticipated nodes for sub-activities, objects (their affordances), and the corresponding spatio-temporal trajectories. We call this augmented graph an anticipatory temporal CRF (ATCRF), formally defined in Section 5.2.

Our goal is to obtain a distribution over the future possibilities, i.e., a distribution over possible ATCRFs. Motivated by particle filtering algorithm [69], we represent this distribution as a set of weighted particles, where each particle is a sampled ATCRF. Partial observations become available as the sub-activity is being performed and we use these partial observations to improve the estimation of the distribution. Section

[3] The skeleton tracker identifies the human hand joints within an error of 10 cm in 82.7% of frames. The SIFT based object tracker gives a bounding box with atleast 10% overlap with the object in 77.8% of the frames.

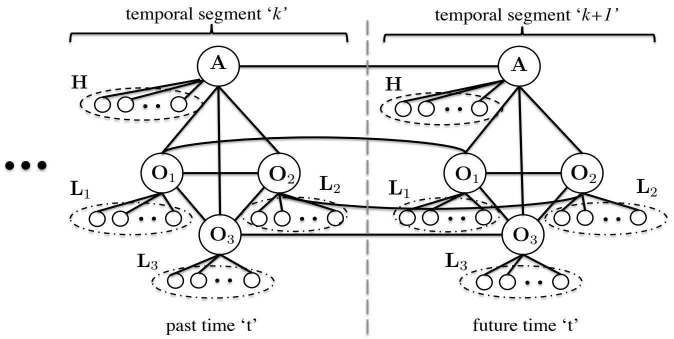


Fig. 3: An ATCRF that models the human poses $\mathcal{H}$, object affordance labels $\mathcal{O}$, object locations $\mathcal{L}$, and sub-activity labels $\mathcal{A}$, over past time 't', and future time 'd'. Two temporal segments are shown in this figure: k^{th} for the recent past, and $(k+1)^{th}$ for the future. Each temporal segment has three objects for illustration in the figure.

5.3 describes this approach. Since each of our ATCRF captures strong context over time (which sub-activity follows another) and space (spatial motion of humans and objects, and their interactions), each of our particles (i.e., possible future) is rich in its modeling capacity. Later, our experiments in Section 8 will show that this is essential for anticipating human actions.

Anticipated temporal segments are generated based on the available object affordances and the current configuration of the 3D scene. For example, if a person has picked up a coffee mug, one possible outcome could be drinking from it. Therefore, for each object, we sample possible locations at the end of the anticipated sub-activity and several trajectories based on the selected affordance. The location and trajectory generation are described in Section 4.1 and Section 4.2 respectively.

The temporal segmentation determines the structure of the ATCRF. It is quite challenging to estimate this structure because of two reasons. First, an activity comprises several sub-activities of varying temporal length, with an ambiguity in the temporal boundaries. Thus a single graph structure may not explain the activity well. Second, there can be several possible graph structures when we are reasoning about activities in the future (i.e., when the goal is to *anticipate* future activities, different from just detecting the past activities). Multiple spatio-temporal graphs are possible in these cases and we need to reason over them in our learning algorithm.

Figure 4 shows two possible graph structures for an activity with two objects. We reason about the possible graph structures for both past and future activities. The key idea is to first sample a few segmentations that are close to the ground-truth using our CRF model instantiated with a subset of features, and then explore the space of segmentation by making merge and split moves to create new segmentations. We do so by approximating the graph with only additive features, which lends to efficient dynamic programming, as described in Section 7.1.

4 Object Affordances

The concept of affordances was proposed by Gibson [10] as all "action possibilities" provided by the environment. Many recent works in computer vision and robotics reason about object functionality (e.g., sittable, drinkable, etc.) instead of object identities (e.g., chairs, mugs, etc.). These works take a recognition based approach to identify the semantic affordance labels [70], [71], [72]. Few recent works explore the physical aspects of affordances based on human interactions [5], [73], [74], [75], [76]. For example, Grabner et al. [73] detect the functionality of the object (specifically, chairs) with respect to possible human poses, Gupta et al. [74] predict stable and feasible human poses given an approximate 3D geometry from an image, etc. In our work, we consider semantic affordances with spatio-temporal grounding which help in anticipating the future activities. Here, we describe how we model the spatio-temporal aspects of affordances.

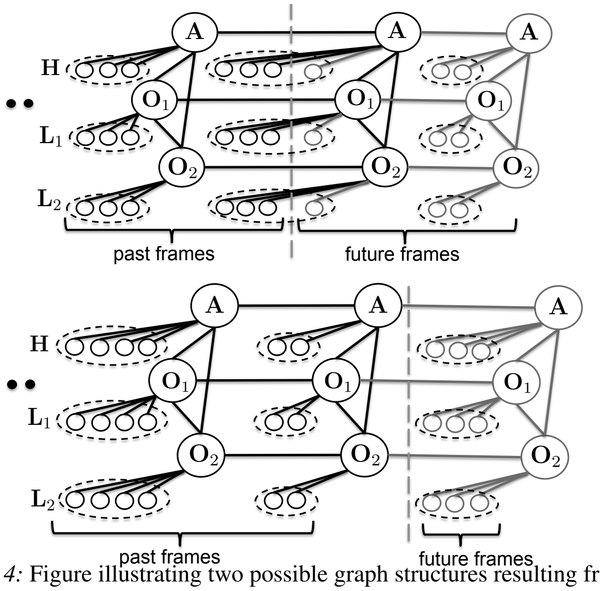


Fig. 4: Figure illustrating two possible graph structures resulting from two temporal segmentations (top and bottom), with six observed frames in the past and three anticipated frames in the future. This example has one sub-activity node and two object nodes in each temporal segment.

4.1 Object Affordance Heatmaps

To represent object affordances we define a potential function based on how the object is being interacted with, when the corresponding affordance is active. The kind of interaction we consider depends on the affordance being considered. For example, when the active affordance of an object is *drinkable*, the object is found near the human's mouth, the interaction considered is the relative position of the object with respect to the human skeleton. In case of the affordance *placeable*, the interaction is the relative position of the object with respect to the environment, i.e., an object is *placeable* when it is above a surface that provides stability to the object once placed. The general form of the potential function for object affordance o given the observations at time t is:

$$\psi_o = \prod_i \psi_{dist_i} \prod_j \psi_{ori_j} \qquad (1)$$

where ψ_{dist_i} is the i^{th} distance potential and ψ_{ori_j} is the j^{th} relative angular potential. We model each distance potential with a Gaussian distribution and each relative angular potential with a von Mises distribution. We find the parameters of the affordance potential functions from the training data using maximum likelihood estimation. Since the potential function is a product of the various components, the parameters of each distribution can be estimated separately. In detail, the mean and variance of the Gaussian distribution have closed form solutions, and we numerically estimate the mean and concentration parameter of the von Mises distribution.

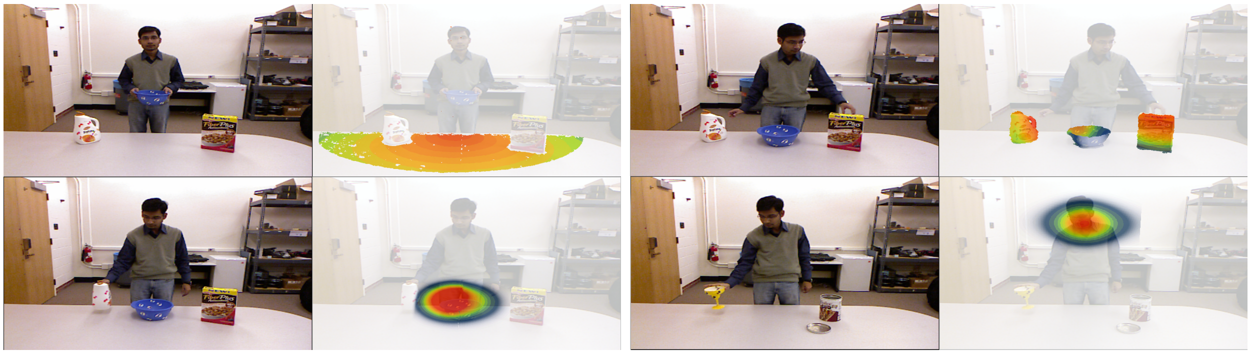


Fig. 5: **Affordance heatmaps**. The learnt affordance heatmaps for *placeability* (top-left), *reachability* (top-right), *pourability* (bottom-left) and *drinkability* (bottom-right). The red signifies where the affordance is most likely, for example, the red signifies where the object is *placeable* (top-left) and the most likely *reachable* locations on the object (top-right) (See Section 4.1).

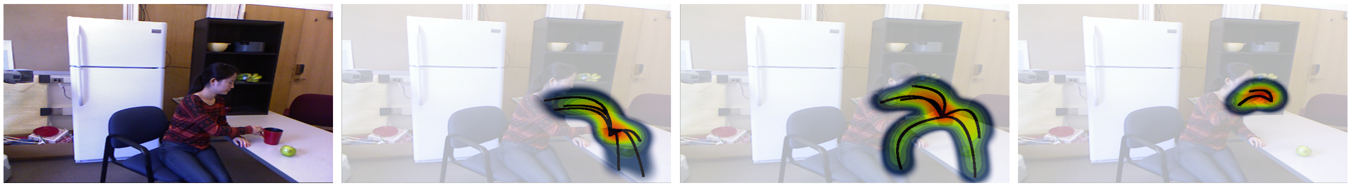


Fig. 6: **Heatmap of anticipated trajectories** for *moving* sub-activity and how they evolve with time.

We categorize these functions into three groups depending on the potentials used: (1) affordances *drinkable* and *reachable* have one distance potential per skeleton joint and one angular potential with respect to the head orientation, (2) affordances depending on the target object, such as *pourable* which depends on a *pour-to* object, have a distance potential and an angular potential with respect to the target object's location, (3) the rest of the affordances which depend on the environment, such *placeable* and *openable*, have a distance potential with respect to the closest surface and an angular potential with respect to the head orientation.

We generate heatmaps for each affordance by scoring the points in the 3D space using the potential function, and the value represents the strength of the particular affordance at that location. Figure 5 shows the heatmaps generated for the *placeable*, *reachable*, *pourable* and *drinkable* affordances. We obtain the future target locations of an object by weighted sampling of the scored 3D points.

4.2 Trajectory Generation

Once a location is sampled from the affordance heatmap, we generate a set of possible trajectories in which the object can be moved form its current location to the predicted target location. We use parametrized cubic equations, in particular Bézier curves, to generate human hand like motions [77].

$$B(x) = (1-x)^3 L_0 + 3(1-x)^2 x L_1 + 3(1-x)x^2 L_2 + x^3 L_3\,, \quad x \in [0,1] \qquad (2)$$

We estimate the control points of the Bézier curves for the proposal distribution component from the trajectories in the training data. A cubic Bézier curve, as shown in Eq. 2, is parameterized by a set of four points: the start and end point of the trajectory (L_0 and L_3 respectively), and two control points (L_1 and L_2) which define the shape of the curve. We first transform and normalize the trajectories in the training data so that all of them have the same start and end points. We then estimate the control points of the Bézier curve, one per sub-activity class, which best fit the normalized trajectories. In detail, L_0 and L_3 are the start and end points of the normalized trajectories, respectively, and L_1 and L_2 are estimated using the least square fitting method to minimize the distance between the fitted and the observed normalized trajectories. Figure 6 shows some of the anticipated trajectories for *moving* sub-activity.

5 Our Approach

Given the observations of a scene containing a human and objects for time t in the past, and its goal is to anticipate future possibilities for time d.

However, for the future d frames, we do not even know the structure of the graph—there may be different number of objects being interacted with depending on which sub-activity is performed in the future. Our goal is to compute a distribution over the possible future states (i.e., sub-activity, human poses and object locations). We will do so by sampling several possible graph structures by augmenting the graph in time, each of which we will call an anticipatory temporal conditional random field (ATCRF). We first describe an ATCRF below.

5.1 Modeling Past with a CRF

MRFs/CRFs are a workhorse of machine learning and have been applied to a variety of applications. Recently, with RGB-D data they have been applied to scene labeling [78] and activity detection [5]. Conditioned on a variety of features as input, the CRFs model rich contextual relations. Learning and inference is tractable in these methods when the label space is discrete and small.

Following [5], we discretize time to the frames of the video[4] and group the frames into temporal segments, where each temporal segment spans a set of contiguous frames corresponding to a single sub-activity. Therefore, at time 't' we have observed 't' frames of the activity that are grouped into 'k' temporal segments. For the past t frames, we know the nodes of the CRF but we do not know the temporal segmentation, i.e., which frame level nodes are connected to

[4] In the following, we will use the number of videos frames as a unit of time, where 1 unit of time ≈ 71ms (=1/14, for a frame-rate of about 14Hz in our experiments).

each of the segment level node. The node labels are also unknown. For a given temporal segmentation, we represent the graph until time t as: $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$, where $\mathcal{E}^t$ represents the edges, and $\mathcal{V}^t$ represents the nodes $\{\mathcal{H}^t, \mathcal{O}^t, \mathcal{L}^t, \mathcal{A}^t\}$: human pose nodes $\mathcal{H}^t$, object affordance nodes $\mathcal{O}^t$, object location nodes $\mathcal{L}^t$, and sub-activity nodes $\mathcal{A}^t$. Figure 3-left part shows the structure of this CRF for an activity with three objects.

Our goal is to model the $P(\mathcal{H}^t, \mathcal{O}^t, \mathcal{L}^t, \mathcal{A}^t | \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}})$, where $\Phi^t_{\mathcal{H}}$ and $\Phi^t_{\mathcal{L}}$ are the observations for the human poses and object locations until time t. Using the independencies expressed over the graph in Figure 3, for a graph $\mathcal{G}^t$, we have:

$$\begin{aligned} &P_{\mathcal{G}^t}(\mathcal{H}^t, \mathcal{O}^t, \mathcal{L}^t, \mathcal{A}^t | \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}}) = \\ &\qquad P(\mathcal{O}^t, \mathcal{A}^t | \mathcal{H}^t, \mathcal{L}^t) P(\mathcal{H}^t, \mathcal{L}^t | \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}}) \quad (3) \end{aligned}$$

The second term $P(\mathcal{H}^t, \mathcal{L}^t | \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}})$ models the distribution of true human pose and object locations (both are continuous trajectories) given the observations from the RGB-D Kinect sensor. We model it using a Gaussian distribution. The first term $P(\mathcal{O}^t, \mathcal{A}^t | \mathcal{H}^t, \mathcal{L}^t)$ predicts the object affordances and the sub-activities that are discrete labels—this term further factorizes following the graph structure as:

$$P(\mathcal{O}^t, \mathcal{A}^t | \mathcal{H}^t, \mathcal{L}^t) \propto \overbrace{\prod_{o_i \in \mathcal{O}} \Psi_{\mathcal{O}}(o_i | \ell_{o_i})}^{\text{object affordance}} \overbrace{\prod_{a_i \in \mathcal{A}} \Psi_{\mathcal{A}}(a_i | h_{a_i})}^{\text{sub-activity}} \prod_{v_i, v_j \in \mathcal{E}} \overbrace{\Psi_{\mathcal{E}}(v_i, v_j | \cdot)}^{\text{edge terms}} \quad (4)$$

Given the continuous state space of $\mathcal{H}$ and $\mathcal{L}$, we rely on [5] for powerful modeling using a discriminative framework for the above term. Each node potential function in Eq. (4), $\Psi_{\mathcal{O}}(o_i | \ell_{o_i})$ and $\Psi_{\mathcal{A}}(a_i | h_{a_i})$, has the form $\sum_{k \in K} y_i^k \left[w_n^k \cdot \phi_n(i) \right]$, where y_i^k denotes a binary variable representing the node i having label k, K is the set of labels, $\phi_n(i)$ is the node feature map and w_n^k are the corresponding node feature weights. Similarly, each edge potential function, $\Psi_{\mathcal{E}}(v_i, v_j | \cdot)$, has the form $\sum_{(l,k) \in K \times K} y_i^l y_j^k \left[w_e^{lk} \cdot \phi_e(i,j) \right]$, where $\phi_e(i,j)$ is the edge feature map and w_e^{lk} are the corresponding edge feature weights. The feature maps, $\phi_n(i)$ and $\phi_e(i,j)$, are described in detail in Section 6. We can rewrite Eq. (4) by taking logarithm on both sides and grouping the node potentials as the following energy function expressed over a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (as illustrated in Figure 2):

$$\begin{aligned} E(\mathbf{y} | \Phi(\mathbf{x}); \mathbf{w}) = &\sum_{i \in \mathcal{V}} \sum_{k \in K} y_i^k \left[w_n^k \cdot \phi_n(i) \right] \\ &+ \sum_{(i,j) \in \mathcal{E}} \sum_{(l,k) \in K \times K} y_i^l y_j^k \left[w_e^{lk} \cdot \phi_e(i,j) \right] \quad (5) \end{aligned}$$

Depending on the nodes and edges in the graph $\mathcal{G}$, the appropriate subset of features and class labels are used in energy function given in Eq. (5). The learning and inference algorithms are described in Section 7.

5.2 Modeling one Possible Future with an augmented temporal CRF (ATCRF).

We defined the anticipatory temporal conditional random field as an augmented graph $\mathcal{G}^{t,d} = (\mathcal{V}^{t,d}, \mathcal{E}^{t,d})$, where t is observed time and d is the future anticipation time. $\mathcal{V}^{t,d} = \{\mathcal{H}^{t,d}, \mathcal{O}^{t,d}, \mathcal{L}^{t,d}, \mathcal{A}^{t,d}\}$ represents the set of nodes in the past time t as well as in the future time d. $\mathcal{E}^{t,d}$ represents the set of all edges in the graph (see Figure 3). The observations (not shown in the figure) are represented as set of features, $\Phi^t_{\mathcal{H}}$ and $\Phi^t_{\mathcal{O}}$, extracted from the t observed video frames. Note that we do not have observations for the future frames.

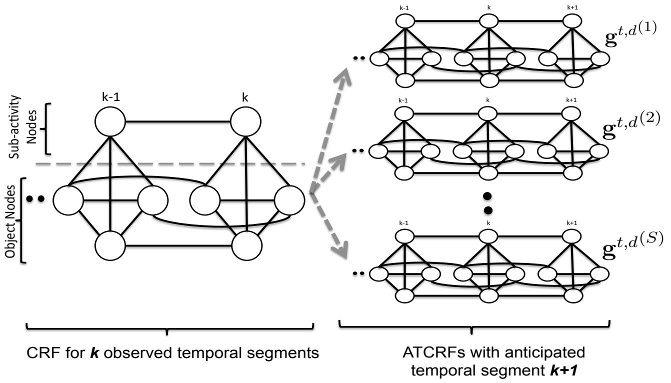


Fig. 7: Figure showing the process of augmenting the CRF structure to obtain multiple ATCRFs at time t for an activity with three objects. The frame level nodes are not shown for the sake of clarity.

In the augmented graph $\mathcal{G}^{t,d}$, we have:

$$\begin{aligned} &P_{\mathcal{G}^{t,d}}(\mathcal{H}^{t,d}, \mathcal{O}^{t,d}, \mathcal{L}^{t,d}, \mathcal{A}^{t,d} | \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}}) = \\ &\qquad P(\mathcal{O}^{t,d}, \mathcal{A}^{t,d} | \mathcal{H}^{t,d}, \mathcal{L}^{t,d}) P(\mathcal{H}^{t,d}, \mathcal{L}^{t,d} | \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}}) \quad (6) \end{aligned}$$

The first term is similar to Eq. (4), except over the augmented graph, and we can still rely on the discriminatively trained CRF presented in [5]. We model the second term with a Gaussian distribution.

5.3 Modeling the Distribution over Future Possibilities with ATCRFs.

There can be several potential augmented graph structures $\mathcal{G}^{t,d}$ because of different possibilities in human pose configurations and object locations that determines the neighborhood graph. Even the number of nodes to be considered in the future changes depending on the sub-activity and the configuration of the environment.

Let $\mathbf{g}^{t,d}$ represent a sample augmented graph structure with particular values assigned to its node variables. I.e., one sample may represent that a person and object move in a certain way, performing a sub-activity with certain object affordances, and another sample may represent a person moving in a different way performing a different sub-activity.

Figure 7 shows the process of augmenting CRF structure corresponding to the seen frames with the sampled anticipations of the future to produce multiple ATCRF particles at time t. The frame level nodes are not shown in the figure. The left portion of the figure shows the nodes corresponding to the k observed temporal segments. This graph is then augmented with a set of anticipated nodes for the temporal segment $k+1$, to generate the ATCRF particles at time t. The frame level nodes of $k+1$ temporal segment are instantiated with anticipated human poses and object locations.

The goal is now to compute the distribution over these ATCRFs $\mathbf{g}^{t,d}$, i.e., given observations until time t, we would like to estimate the posterior distribution $p(\mathbf{g}^{t,d} | \Phi_t)$ from Eq. (6). However, this is extremely challenging because the space of ATCRFs is infinite, and to even represent the distribution we need an exponential number of labels. We therefore represent the posterior using a set of weighted particles as shown in Eq. (7) and choose the weights using importance sampling as shown in Eq. (8).

$$p(\mathbf{g}^{t,d}|\Phi_t) \approx \sum_{s=1}^{S} \hat{w}_t^s \delta_{\mathbf{g}^{t,d(s)}}(\mathbf{g}^{t,d}) \tag{7}$$

$$\hat{w}_t^s \propto \frac{p(\mathbf{g}^{t,d^{(s)}}|\Phi_t)}{q(\mathbf{g}^{t,d^{(s)}}|\Phi_t)} \tag{8}$$

Here, $\delta_x(y)$ is the Kronecker delta function which takes the value 1 if x equals y and 0 otherwise, $\hat{w}_t^s$ is the weight of the sample s after observing t frames, and $q(\mathbf{g}^{t,d}|\Phi_t)$ is the proposal distribution. We need to perform importance sampling because: (a) sampling directly from $p(\mathbf{g}^{t,d}|\Phi_t)$ is not possible because of the form of the distribution in a discriminative framework, and (b) sampling uniformly would be quite naive because of the large space of ATCRFs and most of our samples would entirely miss the likely futures.

We now describe how we sample particles from the proposal distribution $q(\mathbf{g}^{t,d}|\Phi_t)$ and how to evaluate the posterior for the generated samples.

Sampling. In order to generate a particle ATCRF, we need to generate possible human pose and object locations for the d future frames. We write the desired distribution to sample as:

$$\begin{aligned} q(\mathbf{g}^{t,d}|\Phi^t) &= P_{\mathcal{G}^{t,d}}(\mathcal{H}^{t,d}, \mathcal{O}^{t,d}, \mathcal{L}^{t,d}, \mathcal{A}^{t,d}|\Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}}) \\ &= P_{\mathcal{G}^t}(\mathcal{H}^t, \mathcal{O}^t, \mathcal{L}^t, \mathcal{A}^t|\Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}}) \\ &\quad P(\mathcal{H}^d, \mathcal{L}^d|\mathcal{O}^d, \mathcal{A}^d, \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}},)P(\mathcal{O}^d, \mathcal{A}^d|\mathcal{O}^t, \mathcal{A}^t, \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}}) \end{aligned} \tag{9}$$

We first sample the affordances, one per object in the scene, and the corresponding sub-activity from the distribution $P(\mathcal{O}^d, \mathcal{A}^d|\Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}})$. This is a discrete distribution generated from the training data based on the object type (e.g., cup, bowl, etc.) and object's current position with respect to the human in the scene (i.e., in contact with the hand or not). For example, if a human is holding an object of type 'cup' placed on a table, then the affordances *drinkable* and *movable* with their corresponding sub-activities (*drinking* and *moving* respectively) have equal probability, with all others being 0.[5]

Once we have the sampled affordances and sub-activity, we need to sample the corresponding object locations and human poses for the d anticipated frames from the distribution $P(\mathcal{H}^d, \mathcal{L}^d|\mathcal{O}^d, \mathcal{A}^d, \Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}})$. In order to have meaningful object locations and human poses we take the following approach. We sample a set of target locations and motion trajectory curves based on the sampled affordance, sub-activity and available observations. We then generate the corresponding object locations and human poses from the sampled end point and trajectory curve. The details of sampling the target object location and motion trajectory curves are described in Section 4.1 and Section 4.2 respectively.

Scoring. Once we have the sampled ATCRF particles, we obtain the weight of each sample s by evaluating the posterior for the given sample, $q(\mathbf{g}^{t,d^{(s)}}|\Phi^t)$, as shown in Eq. (9) and normalize the weights across the samples.

Note that the aforementioned methods for the affordance and trajectory generation are only for the proposal distribution to sample. The estimated trajectories are finally scored using our ATCRF model.

6 Features: Additive and Non-Additive

In this section we describe the feature maps used in the energy function Eq. 5. In detail, we use the same features as described by KGS [5]. These features include the node feature maps $\phi_o(i)$ and $\phi_a(j)$ for object node i and sub-activity node j respectively, and edge feature maps $\phi_e(i,j)$ capturing the relations between various nodes. The object node feature map, $\phi_o(i)$, includes the (x, y, z) coordinates of the object's centroid, the coordinates of the object's bounding box and transformation matrix w.r.t. to the previous frame computed at the middle frame of the temporal segment, the total displacement and distance moved by the object's centroid in the set of frames belonging to the temporal segment. The sub-activity node feature map, $\phi_a(j)$, gives a vector of features computed using the noisy human skeleton poses obtained from running Openni's skeleton tracker on the RGBD video. We compute the above described location (relative to the subject's head location) and distance features for each the upper-skeleton joints excluding the elbow joints (neck, torso, left shoulder, left palm, right shoulder and right palm).

The edge feature maps, $\phi_t(i,j)$, include relative geometric features such as the difference in (x, y, z) coordinates of the object centroids and skeleton joint locations and the distance between them. In addition to computing these values at the first, middle and last frames of the temporal segment, we also consider the min and max of their values across all frames in the temporal segment to capture the relative motion information. The temporal relational features capture the change across temporal segments and we use the vertical change in position and the distance between corresponding object and joint locations. We perform cumulative binning of all the feature values into 10 bins for each feature.

We categorize the features into two sets: additive features, $\Phi^A(\mathbf{x})$, and non-additive features, $\Phi^{NA}(\mathbf{x})$. We compute the additive features for a set of frames corresponding to a temporal segment by adding the feature values for the frames belonging to the temporal segment. Examples of the additive features include distance moved and vertical displacement of an object within a temporal segment. The features that do not satisfy this property are referred to as the non-additive features, for example, maximum and minimum distances between two objects. As we discuss in the next section, additive features allow efficient joint segmentation and labeling by using dynamic programming, but may not be expressive enough.

Non-additive features sometimes provide very useful cues for discriminating the sub-activity and affordance classes. For example, consider discriminating *cleaning* sub-activity from a *moving* sub-activity: here the total distance moved could be similar (an additive feature), however, the minimum and maximum distance moved being small may be strong indicator of the activity being *cleaning*. In fact, when compared to our model learned using only the additive features, the model learned with both additive and non-additive features improves macro precision and recall by 5% and 10.1% for labeling object affordance respectively and by 3.7% and 6.2% for labeling sub-activities respectively.

[5] If we know the object type as well, then it helps in generating informative samples based on the discrete distribution learnt from the training data, which can save computation time as fewer samples are sufficient. In our experiments, when object type information is not provided, the anticipation performance (micro P/R) for sub-activities and affordances only drops by a maximum of 0.3% for our full model (Table 2-row 6).

7 Learning and Inference

7.1 Inference

The inference problem is to find the best labeling of the past nodes of the CRF for detecting the past sub-activities and affordances and past as well as augmented future nodes of the ATCRF for anticipation. The prediction $\hat{\mathbf{y}}$ is computed as the argmax of an energy function $E(\mathbf{y}|\Phi(\mathbf{x});\mathbf{w})$.

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} E(\mathbf{y}|\Phi(\mathbf{x});\mathbf{w}) \tag{10}$$

For a given temporal segmentation, where the graph structure is fully known, finding the argmax over labelings is a NP hard problem. However, its equivalent formulation as the following mixed-integer program has a linear relaxation which can be solved efficiently as a quadratic pseudo-Boolean optimization problem using a graph-cut method [79] as described in [5].

$$\begin{aligned}\hat{\mathbf{y}} =& \operatorname*{argmax}_{\mathbf{y}} \max_{\mathbf{z}} \sum_{i\in\mathcal{V}} \sum_{k\in K} y_i^k \left[w_n^k \cdot \phi_n(i)\right] \\ &+ \sum_{(i,j)\in\mathcal{E}} \sum_{(l,k)\in K\times K} z_{ij}^{lk} \left[w_e^{lk} \cdot \phi_e(i,j)\right] \qquad (11)\end{aligned}$$

$$\forall i,j,l,k:\ z_{ij}^{lk} \le y_i^l,\ z_{ij}^{lk} \le y_j^k,\ y_i^l + y_j^k \le z_{ij}^{lk} + 1,\ z_{ij}^{lk}, y_i^l \in \{0,1\} \tag{12}$$

Note that the products $y_i^l y_j^k$ have been replaced by auxiliary variables z_{ij}^{lk}. Relaxing the variables z_{ij}^{lk} and y_i^l to the interval $[0,1]$ results in a linear program that can be shown to always have half-integral solutions (i.e. y_i^l only take values $\{0, 0.5, 1\}$ at the solution) [80]. When we consider the additional constraints that each node can take only one label, the problem can no longer be solved via graph cuts. We compute the exact mixed integer solution including these additional constraints using a general-purpose MIP solver[6] during inference.

However, during inference we only know the nodes in the graph but not the temporal segmentation, i.e., the structure of the graph in terms of the edges connecting frame level nodes to the segment level label nodes. We could search for the best labeling over *all* possible segmentations, but this is very intractable because our feature maps contain non-additive features (that are important as described in Section 6).

Efficient Inference with Additive Features. We express the feature set, $\Phi(\mathbf{x})$, as the concatenation of the additive and non-additive feature sets, $\Phi^A(\mathbf{x})$ and $\Phi^{NA}(\mathbf{x})$ respectively. Therefore, by rearranging the terms in Eq. (4), the energy function can be written as:

$$E(\mathbf{y}|\Phi(\mathbf{x});\mathbf{w}) = E(\mathbf{y}|\Phi^A(\mathbf{x});\mathbf{w}) + E(\mathbf{y}|\Phi^{NA}(\mathbf{x});\mathbf{w})$$

We perform efficient inference for the energy term $E(\mathbf{y}|\Phi^A(\mathbf{x});\mathbf{w})$ by formulating it as a dynamic program (see Eq. (13)). In detail, let L denote the max length of a temporal segment, i denote the frame index, s denote the temporal segment spanning frames $(i-l)$ to i, and $(s-1)$ denote the previous segment. We write the energy function in a recursive form as:

$$\begin{aligned}V(i,k) =& \max_{k',l=1\ldots L} V(i-l,k') + \sum_{k\in K} y_s^k \left[w_n^k \cdot \phi_n^A(s)\right] \\ &+ \sum_{k\in K} y_s^k \left[w_e^{lk} \cdot \phi_e^A(s-1,s)\right] \qquad (13)\end{aligned}$$

Here, $\phi_n^A(s)$ and $\phi_e^A(s-1,s)$ denote the additive feature maps and can be efficiently computed by using the concept of integral images.[7] The best segmentation then corresponds to the path traced by $\max_a V(t,a)$, where t is the number of video frames. Using $E(\mathbf{y}|\Phi^A(\mathbf{x});\mathbf{w})$, we find the top-k scored segmentations[8] and then evaluate them using the full model $E(\mathbf{y}|\Phi(\mathbf{x});\mathbf{w})$ in order to obtain more accurate labelings.

Merge and Split Moves. The segmentations generated by the approximate energy function, $E(\mathbf{y}|\Phi^A(\mathbf{x});\mathbf{w})$, are often very close to the given ground-truth segmentations. However, since the energy function used is only approximate, it sometimes tends to over-segment or miss the boundary by a few frames. In order to obtain a representative set of segmentation samples, we also perform random merge and split moves over these segmentations, and consider them for evaluating with the full model as well. A merge move randomly selects a boundary and removes it, and a split move randomly chooses a frame in a segment and creates a boundary.

Heuristic Segmentations. There is a lot of information present in the video which can be utilized for the purpose of temporal segmentation. For example, smooth movement of the skeleton joints usually represent a single sub-activity and the sudden changes in the direction or speed of motion indicate sub-activity boundaries. Therefore, we incorporate such information in performing temporal segmentation of the activities. In detail, we use the multiple segmentation hypotheses proposed by KGS. These include graph based segmentation method proposed by [81] adapted to temporally segment the videos. The sum of the Euclidean distances between the skeleton joints and the rate of change of the Euclidean distance are used as the edge weights for two heuristic segmentations respectively. By varying the thresholds, different temporal segmentations of the given activity can be obtained. In addition to the graph based segmentation methods, we also use the uniform segmentation method which considers a set of continuous frames of fixed size as the temporal segment. There are two parameters for this method: the segment size and the offset (the size of the first segment). However, these methods often over-segment a sub-activity, and each segmentation would result in a different graph structure for our CRF modeling.

We generate multiple graph structures for various values of the parameters for the above mentioned methods and obtain the predicted labels for each using Eq. (10). We obtain the final labeling over the segments by either using the second-step learning method presented in KGS, or by performing voting and taking the label predicted by majority of the sampled graph structures (our experiments in Section 8.2 follow the latter). During anticipation, we only consider the best graph structure obtained form the using the additive features and split-merge moves. Algorithm 1 gives the summary of the inference algorithm for anticipation.

[6] http://www.tfinley.net/software/pyglpk/readme.html

[7] The additive features for temporal segments starting at the first frame and ending at frame l, for $l = 1..t$ are precomputed, i.e., the segment features for a total of t temporal segments are computed. This needs $(n \times t)$ summations, where n is the number of features. Now the segment features for a temporal temporal segment starting and ending at any frame can be computed by n subtractions. Therefore, the total feature computation cost is linear in the number of possible segmentations.

[8] k is 2 in experiments.

```
Data: RGB-D video frames
Result: Future sub-activity and affordance anticipations
$t = 0$, $P = \{\}$ ;
while new frame $f_t$ observed do
    Generate frame features for frame $f_t$ (Section 6);
    if temporal segmentation not given then
        Find best segmentation using additive energy
        $E(\mathbf{y}|\Phi^A(\mathbf{x});\mathbf{w})$ (solve Eq. 13);
        Sample segmentations by split and merge moves
        (Section 7.1);
    end
    Compute segment features $\phi_n$ and $\phi_e$ (Section 6);
    Compute $\hat{\mathbf{y}}$, best labeling of the past-CRF (Eq. 11-12);
    for each object do
        Sample possible future affordance and sub-activity
        from the discrete distribution $P(\mathcal{O}^d, \mathcal{A}^d|\Phi^t_{\mathcal{H}}, \Phi^t_{\mathcal{L}})$;
        Sample future object location based on the affordance
        heatmaps $\psi_o$;
        Generate corresponding object trajectory and human
        poses for $d$ future frames;
        Augment the past-CRF to generate an ATCRF particle
        $\mathbf{g}^{t,d^{(s)}}$ ;
        $P = P \cup \{\mathbf{g}^{t,d^{(s)}}\}$;
    end
    for each particle $\mathbf{g}^{t,d^{(s)}} \in P$ do
        for each augmented frame do
            Generate frame features (Section 6);
        end
        if temporal segmentation not given then
            Find best segmentation using additive energy
            $E(\mathbf{y}|\Phi^A(\mathbf{x});\mathbf{w})$ (solve Eq. 13);
            Sample segmentations by split and merge moves
            (Section 7.1);
        end
        Compute segment features $\phi_n$ and $\phi_e$ (Section 6);
        Compute $\hat{\mathbf{y}}$, best labeling for the ATCRF particle
        (Eq. 11-12);
        Compute weight $\hat{w}^s_t$ (Eq. 8);
    end
    $P$ = top-k scored particles in $P$;
    $A_t$ = future sub-activity and affordance labels of top-3
    particles based on $E(\mathbf{y}|\Phi(\mathbf{x});\mathbf{w})$;
    $t = t + 1$;
end
```

Algorithm 1: Summary of our inference method.

7.2 Learning

The structure of the graph is fully known during learning. We obtain the parameters of the energy function in Eq. (5) by using the cutting plane method [82] as described in [5]. Given M labeled training examples $(\mathbf{x}_1, \mathbf{y}_1), .., (\mathbf{x}_M, \mathbf{y}_M)$, we optimize the regularized upper bound on the training error.

$$R(h) = \tfrac{1}{M}\textstyle\sum_{m=1}^{M} \Delta(\mathbf{y}_m, \hat{\mathbf{y}}_m),$$

where $\hat{\mathbf{y}}_m$ is the optimal solution of Eq. (10) and $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ is the loss function defined as

$$\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \textstyle\sum_{i\in\mathcal{V}_o}\sum_{k\in K_o} |y_i^k - \hat{y}_i^k| + \sum_{i\in\mathcal{V}_a}\sum_{k\in K_a} |y_i^k - \hat{y}_i^k|.$$

To simplify notation, note that Eq. (11) can be equivalently written as $\mathbf{w}^T\Gamma(\mathbf{x}, \mathbf{y})$ by appropriately stacking the w_n^k and w_t^{lk} into $\mathbf{w}$ and the $y_i^k\phi_a(i)$, $y_i^k\phi_o(i)$ and $z_{ij}^{lk}\phi_t(i,j)$ into $\Gamma(\mathbf{x}, \mathbf{y})$, where each z_{ij}^{lk} is consistent with Eq. (12) given $\mathbf{y}$. Training can then be formulated as the following convex quadratic program [82]:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\xi \qquad\qquad (14)\\ s.t. \quad & \forall \bar{\mathbf{y}}_1, ..., \bar{\mathbf{y}}_M \in \{0, 0.5, 1\}^{N\cdot K} : \\ & \frac{1}{M}\mathbf{w}^T \sum_{m=1}^{M} [\Gamma(\mathbf{x}_m, \mathbf{y}_m) - \Gamma(\mathbf{x}_m, \bar{\mathbf{y}}_m)] \geq \Delta(\mathbf{y}_m, \bar{\mathbf{y}}_m) - \xi, \end{aligned}$$

where N is the number of temporal segments per example and K is the total number class labels. While the number of constraints in this QP is exponential in M, N and K, it can nevertheless be solved efficiently using the cutting-plane algorithm [82]. The algorithm needs access to an efficient method for computing

$$\bar{\mathbf{y}}_m = \underset{\mathbf{y}\in\{0,0.5,1\}^{N\cdot K}}{\operatorname{argmax}} \left[\mathbf{w}^T\Psi(\mathbf{x}_m, \mathbf{y}) + \Delta(\mathbf{y}_m, \mathbf{y})\right]. \qquad (15)$$

Due to the structure of $\Delta(.,.)$, this problem is identical to the relaxed prediction problem in Eqs. (11)-(12) and can be solved efficiently using graph cuts.

8 Experiments

In this section we describe the detailed evaluation of our approach on both offline data as well as realtime robotic experiments. We first give the details of the dataset in Section 8.1. We then present the detection results in Section 8.2, anticipation results in Section 8.3 and the robotic experiments for anticipation in Section 8.4.

8.1 Data

We use CAD-120 dataset [5], which has 120 RGB-D videos of four different subjects performing 10 high-level activities. The data is annotated with object affordance and sub-activity labels and includes ground-truth object categories, tracked object bounding boxes and human skeletons. The set of high-level activities are: {*making cereal*, *taking medicine*, *stacking objects*, *unstacking objects*, *microwaving food*, *picking objects*, *cleaning objects*, *taking food*, *arranging objects*, *having a meal*}, the set of sub-activity labels are: {*reaching*, *moving*, *pouring*, *eating*, *drinking*, *opening*, *placing*, *closing*, *scrubbing*, *null*} and the set of affordance labels are: {*reachable*, *movable*, *pourable*, *pourto*, *containable*, *drinkable*, *openable*, *placeable*, *closable*, *scrubbable*, *scrubber*, *stationary*}. We use all sub-activity classes for prediction of observed frames but do not anticipate *null* sub-activity.

8.2 Detection Results

For comparison, we follow the same train-test split described in KGS [5] and train our model on activities performed by three subjects and test on activities of a *new subject*. We report the results obtained by 4-fold cross validation by averaging across the folds. We consider the overall micro accuracy (P/R), macro precision and macro recall of the detected sub-activities, affordances and overall activity. Micro accuracy is the percentage of correctly classified labels. Macro precision and recall are the averages of precision and recall respectively for all classes.

Table 1 shows the performance of our proposed approach on object affordance, sub-activity and high-level activity labeling for past activities. Rows 3-5 show the performance for the case where ground-truth temporal segmentation is provided and rows 6-9 show the performance for the different methods when no temporal segmentation is provided. With known

TABLE 1: **Results on CAD-120 dataset for *detection***, showing average micro precision/recall, and average macro precision and recall for affordances, sub-activities and high-level activities. Computed from 4-fold cross validation with testing on a new human subject in each fold. Standard error is also reported.

With ground-truth segmentation.

	Object Affordance			Sub-activity			High-level Activity		
	micro	macro		micro	macro		micro	macro	
method	P/R	Prec.	Recall	P/R	Prec.	Recall	P/R	Prec.	Recall
chance	8.3 (0.0)	8.3 (0.0)	8.3 (0.0)	10.0 (0.0)	10.0 (0.0)	10.0 (0.0)	10.0 (0.0)	10.0 (0.0)	10.0 (0.0)
max class	65.7 (1.0)	65.7 (1.0)	8.3 (0.0)	29.2 (0.2)	29.2 (0.2)	10.0 (0.0)	10.0 (0.0)	10.0 (0.0)	10.0 (0.0)
KGS [5]	91.8 (0.4)	**90.4** (2.5)	74.2 (3.1)	86.0 (0.9)	84.2 (1.3)	76.9 (2.6)	84.7 (2.4)	85.3 (2.0)	84.2 (2.5)
Our model: all features	**93.9** (0.4)	89.2 (1.3)	**82.5** (2.0)	**89.3** (0.9)	**87.9** (1.8)	**84.9** (1.5)	**93.5** (3.0)	**95.0** (2.3)	**93.3** (3.1)
Our model: only additive features	92.0 (0.5)	84.2 (2.2)	72.4 (1.2)	86.5 (0.6)	84.2 (1.3)	78.7 (1.9)	90.3 (3.8)	92.8 (2.7)	90.0 (3.9)
Without ground-truth segmentation.									
Our DP seg.	83.6 (1.1)	70.5 (2.3)	53.6 (4.0)	**71.5** (1.4)	71.0 (3.2)	60.1 (3.7)	80.6 (4.1)	86.1 (2.5)	80.0 (4.2)
Our DP seg. + moves	84.2 (0.9)	72.6 (2.3)	58.4 (5.3)	71.2 (1.1)	70.6 (3.7)	61.5 (4.5)	83.1 (5.2)	**88.0** (3.4)	**82.5** (5.4)
heuristic seg. (KGS)	83.9 (1.5)	75.9 (4.6)	64.2 (4.0)	68.2 (0.3)	71.1 (1.9)	62.2 (4.1)	80.6 (1.1)	81.8 (2.2)	80.0 (1.2)
Our DP seg. + moves + heuristic seg.	**85.4** (0.7)	**77.0** (2.9)	**67.4** (3.3)	70.3 (0.6)	**74.8** (1.6)	**66.2** (3.4)	83.1 (3.0)	87.0 (3.6)	82.7 (3.1)

TABLE 2: **Anticipation Results of Future Activities and Affordances**, computed over 3 seconds in the future (similar trends hold for other anticipation times).

	Anticipated Sub-activity			Anticipated Object Affordance		
model	micro P/R	macro F1-score	robot anticipation metric	micro P/R	marco F1-score	robot anticipation metric
chance	10.0 ± 0.1	10.0 ± 0.1	30.0 ± 0.1	8.3 ± 0.1	8.3 ± 0.1	24.9 ± 0.1
Nearest-neighbor	22.0 ± 0.9	10.6 ± 0.6	48.1 ± 0.5	48.3 ± 1.5	17.2 ± 1.0	60.9 ± 1.1
KGS [5] + co-occurence	28.6 ± 1.8	11.1 ± 0.4	34.6 ± 2.8	55.9 ± 1.7	11.6 ± 0.4	62.0 ± 1.8
ATCRF-discrete	34.3 ± 0.8	12.2 ± 0.2	44.8 ± 1.1	59.5 ± 1.5	12.4 ± 0.3	67.6 ± 1.3
ATCRF	47.7 ± 1.6	37.9 ± 2.6	69.2 ± 2.1	66.1 ± 1.9	36.7 ± 2.3	71.3 ± 1.7
Ours-full	**49.6** ± 1.4	**40.6** ± 1.6	**74.4** ± 1.6	**67.2** ± 1.1	**41.4** ± 1.5	**73.2** ± 1.0

graph structure, the model using the the full set of features (row 4) outperforms the model which uses only the additive features (row 5): macro precision and recall improve by 5% and 10.1% for labeling object affordance respectively and by 3.7% and 6.2% for labeling sub-activities respectively. This shows that additive features bring us close, but not quite, to the optimal graph structure.

When the graph structure is not known, the performance drops significantly. Our graph sampling approach based on the additive energy function (row 6) achieves 83.6% and 71.5% micro precision for labeling object affordance and sub-activities, respectively. This is improved by sampling additional graph structures based on the Split and Merge moves (row 7). Finally, combining these segmentations with the other heuristically generated segmentations presented by KGS, our method obtains the best performance (row 9) and significantly improves the previous state-of-the-art (KGS, row 8).

Figure 8 shows the confusion matrix for labeling affordances, sub-activities and high-level activities using our method (row 9). Note that there is a strong diagonal with a few errors such as *pouring* misclassified as *moving*, and *picking objects* misclassified as *having a meal*. Figure 9 shows the labeling output of the different methods. The bottom-most row show the ground-truth segmentation, top-most row is the labeling obtained when the graph structure is provided, followed by three heuristically generated segmentations. The fifth row is the segmentation generated by our sampling approach and the sixth and seventh rows are the labeling obtained by combining the multiple segmentations using a simple max-voting and by the multi-segmentation learning of KGS. Note that some sub-activity boundaries are more ambiguous (high variance among different methods) than the others. Our method has an end-to-end (including feature computation cost) frame rate of 4.3 frames/sec compared to 16.0 frames/sec of KGS.

8.3 Anticipation Results

Baseline Algorithms. We compare our method against the following baselines: *1) Chance.* The anticipated sub-activity and affordance labels are chosen at random.

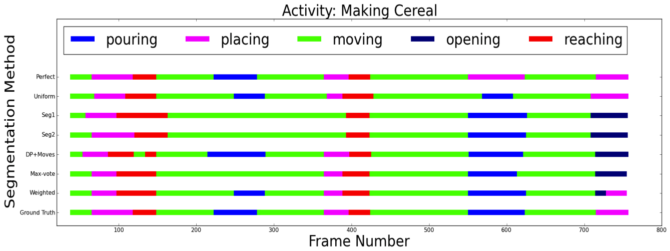


Fig. 9: **Illustration of the ambiguity in temporal segmentation.** We compare the sub-activity labeling of various segmentations. Here, *making cereal* activity comprises the sub-activities: *reaching*, *moving*, *pouring* and *placing* as colored in red, green, blue and magenta respectively. The x-axis denotes the time axis numbered with frame numbers. It can be seen that the various individual segmentation methods are not perfect.

2) Nearest Neighbor Exemplar. It first finds an example from the training data which is the most similar to the activity observed in the last temporal segment. The sub-activity and object affordance labels of the frames following the matched frames from the exemplar are predicted as the anticipations. To find the exemplar, we perform a nearest neighbor search in the feature space for the set of frames, using the node features described in KGS [5].

3) Co-occurrence Method. The transition probabilities for sub-activities and affordances are computed from the training data. The observed frames are first labelled using the MRF model from KGS. The anticipated sub-activity and affordances for the future frames are predicted based on the transition probabilities given the inferred labeling of the last frame.

4) ATCRF without $\{\mathcal{H}, \mathcal{L}\}$ anticipation (ATCRF-discrete). Our ATCRF model with only augmented nodes for discrete labels (sub-activities and object affordances).

5) ATCRF. Our method that samples the future nodes (both segment and frame level) as described in Section 5.3, and uses a fixed temporal structure, which in this case is the segmentation output of KGS.

Evaluation: We follow the same train-test split described in KGS and train our model on activities performed by three subjects and test on activities of a *new subject*. We report the

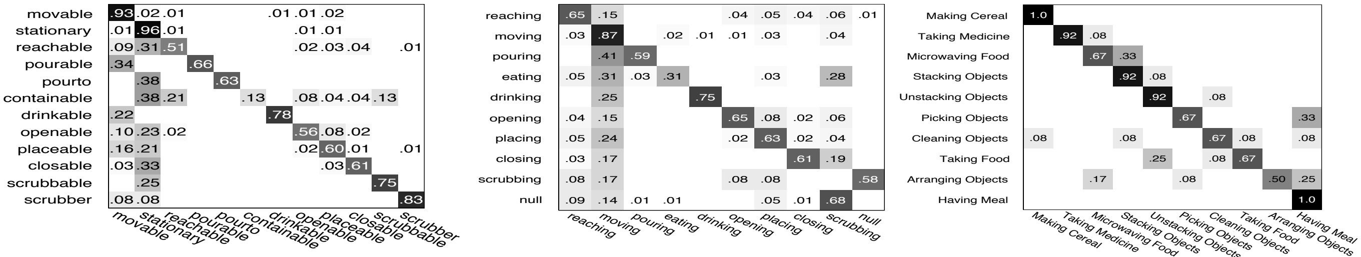


Fig. 8: **Confusion matrix** for affordance labeling (left), sub-activity labeling (middle) and high-level activity labeling (right) of the test RGB-D videos.

results obtained by 4-fold cross validation by averaging across the folds. We consider the following metrics:

1) Labeling Metrics. For detecting and anticipating labels (for sub-activity and affordances), we compute the overall micro accuracy (P/R), macro precision, macro recall and macro F1 score. Micro accuracy is the percentage of correctly classified labels. Macro precision and recall are the averages of precision and recall respectively for all classes.

2) Robot Anticipation Metric. It is important for a robot to plan ahead for multiple future activity outcomes. Therefore, we measure the accuracy of the anticipation task for the top three predictions of the future. If the actual activity matches one of the top three predictions, then it counts towards positive.

3) Trajectory Metric. For evaluating the quality of anticipating trajectories, we compute the modified Hausdorff distance (MHD) as a physical measure of the distance between the anticipated object motion trajectories and the true object trajectory from the test data.[9]

Table 2 shows the frame-level metrics for anticipating sub-activity and object affordance labels for 3 seconds in the future on the CAD-120 dataset. We use the temporal segmentation algorithm from KGS for obtaining the graph structure of the observed past frames for all the baseline methods. ATCRF (row 5) outperforms all the baseline algorithms and achieves a significant increase across all metrics. Our full model (row 6), which estimates the graph structure for both past and the future, improves the anticipation performance further. Figure 10 shows the highest scored anticipations for the *cleaning objects* activity. We will now study our results on anticipation in the form of the following questions:

How does the performance change with the duration of the future anticipation? Figure 11 shows how the macro F1 score and the *robot anticipation metric* changes with the anticipation time. The average duration of a sub-activity in the CAD-120 dataset is around 3.6 seconds, therefore, an anticipation duration of 10 seconds is over two to three sub-activities. With the increase in anticipation duration, performance of the others approach that of a random chance baseline, the performance of our ATCRF declines. It still outperforms other baselines for all anticipation times.

How does the performance change with the duration of the past observations? Figure 12 shows how the macro F1 score changes with the past observation time and future anticipation time. The algorithm has lower performance when predicting longer into the future, but this improves as more observations become available. Therefore, context from the past helps in anticipating longer into the future.

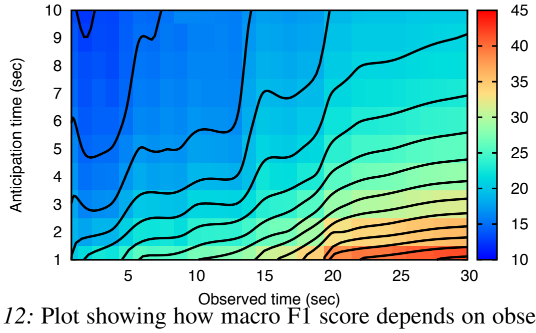


Fig. 12: Plot showing how macro F1 score depends on observed time and anticipation time.

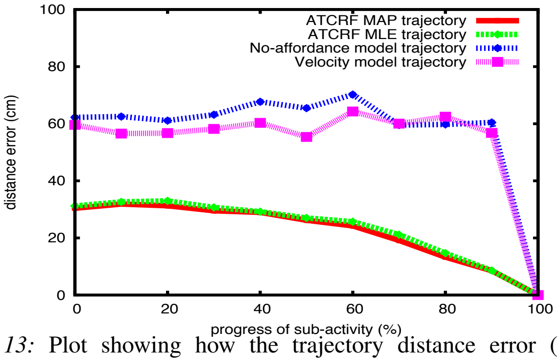


Fig. 13: Plot showing how the trajectory distance error (MHD) changes with the progress of the activity for our ATCRF (top particle and local mean) and other baselines (Kalman Filter velocity model using object affordance as target, and one without object affordance information).

How good are the anticipated trajectories? Since trajectories are continuous variables, we perform two types of estimation: MAP, where we take the highest scored particle generated by our model, and MLE where we take the weighted sum. Figure 13 shows how these distance errors, averaged over all the moving sub-activities in the dataset, change with the progress of the sub-activity. Figure 6 shows the sampled trajectories along with the heatmap corresponding to the distribution of trajectories. At the beginning of the sub-activity the anticipations correspond to moving the cup to other places on the table and near the mouth to drink. As the sub-activity progresses, depending on the current position of the cup, a few new target locations become probable, such as moving the cup on to the lap (such instances are observed in the training data). These new possibilities tend to increase the distance measure as can be seen in the plot of Figure 13. However, on observing more frames, the intent of the human is inferred more accurately resulting in better anticipated trajectories, for example in Figure 6-last frame, anticipating only moving to

[9] The MHD allows for local time warping by finding the best local point correspondence over a small temporal window. When the temporal window is zero, the MHD is same as the Euclidean distance between the trajectories. We normalize the distance by the length of the trajectory in order to compare across trajectories of different lengths. The units of the MHD are centimeters.

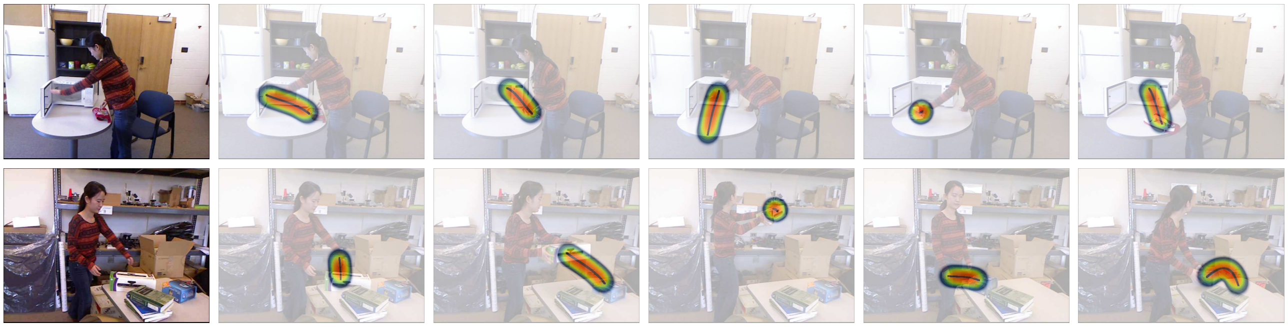


Fig. 10: **Highest scored future anticipations** for *cleaning objects* activity (top-row) and *arranging objects* activity (bottom-row).

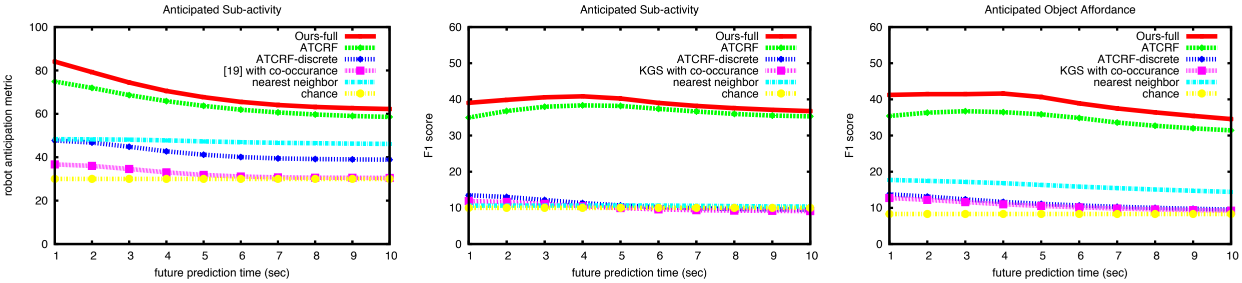


Fig. 11: Plots showing how *robot anticipation metric* and macro F1 score changes with the future anticipation time for all methods.

TABLE 3: **Online Detection Results** of Past Activities and Affordances.

model	Past Sub-activity Detection			Past Object Affordance Detection		
	micro	macro		micro	macro	
	P/R	Prec.	Recall	P/R	Prec.	Recall
chance	10.0 (0.1)	10.0 (0.1)	10.0 (0.1)	8.3 (0.1)	8.3 (0.1)	8.3 (0.1)
KGS [5] - online	80.3 (1.5)	78.0 (1.3)	68.1 (2.6)	89.6 (0.8)	80.7 (2.8)	67.8 (1.4)
ATCRF-discrete	84.0 (1.3)	72.2 (2.3)	60.7 (2.3)	87.7 (1.0)	67.9 (2.4)	48.9 (2.6)
ATCRF	**84.7** (1.4)	**80.6** (1.0)	**75.6** (2.4)	**92.3** (0.7)	**84.8** (2.3)	**77.1** (1.1)

drink trajectories.

Effect of anticipation on detection of past activities. Table 3 shows the detection results of the sub-activities and object affordances of the past temporal segments, computed in an online fashion. When we label each past segment, we observe that segment's features but not the future. The online metrics are computed by aggregating performance on the recent past of three segments. (KGS's method was to label a segment given past, present, as well as the future.) In this experiment, we assumed ground-truth segmentation and object tracks for consistent comparison across the methods. If we instead use an algorithm to segment [5], the overall performance drops, however similar trends hold. We see that both the anticipation methods (rows 3-4) improve the detection results over the one that does not anticipate (row 2). This shows that anticipating the future can improve present and past performance on detection.

Effect of unknown activities on anticipation performance. For activities not present in the training data, our approach generates most likely anticipations based on the affordances of the objects present in the environment and the detected past sub-activities. However, our approach cannot anticipate a sub-activity on which the anticipation model was not trained. In fact, around 14% of the segments in our dataset are sub-activities which do not belong to the nine sub-activity categories used to train our anticipation model. These sub-activities include various background actions such as *checking time*, *standing still*, etc., which are not relevant to the high-level activity being performed. We label these segments as *null* sub-activities and include them for learning the energy function in Eq. 4. Therefore, even though our model is unable to anticipate such sub-activities in the future, they are correctly detected as *null* sub-activities. This allows us to ignore such irrelevant sub-activities and proceed to anticipate the most likely future. However, if *null* sub-activities are performed very often, the performance of our anticipation model would go down further.

8.4 Robotic Experiments

In this section we show how future activity predictions can help the robot perform appropriate actions in response to what the human is going to do next. By incorporating such reactive responses, the robot can better assist humans in tasks which they are unable to perform as well as work along side the humans much more efficiently.

We use a PR2 robot to demonstrate the following anticipatory response scenarios: (i) Robot is instructed to refill water glasses for people seated at a table, but when it sees a person reaching a glass to drink, it waits for him to finish drinking before refilling, in order to avoid spilling, and (ii) Robot opens the fridge door when a person approaches the fridge to place something inside the fridge. PR2 is mounted with a Kinect as its main input sensor to obtain the RGB-D video stream. We used the OpenRAVE libraries [83] for programing the robot to perform the pre-programmed tasks described in the aforementioned scenarios by incorporating the anticipations generated with our ATCRFs. Figure 1 and Figure 14 show the snapshots of the robot observing the human, the anticipated actions and the response executed by the robot.

In our experiments, on the first scenario, we evaluate the success rate which is defined as the percentage of times the robot identifies the correct response. We have a new subject (not seen in the training data) performing the interaction task multiple times in addition to other activities which should not effect the robot's response, such as reaching for a book, etc. We considered a total of 10 interaction tasks which involve four objects including the cup, and 5 of these tasks were to

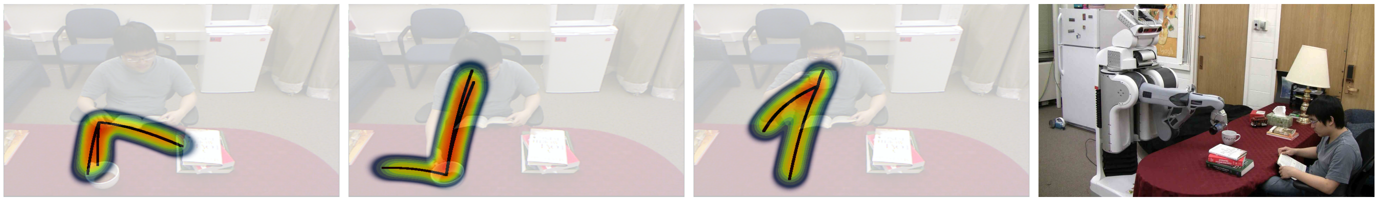


Fig. 14: **Robot Anticipatory Response** for refilling water task. See Figure 1 for opening fridge door task.

reach for the cup and drink from it. The robot is given an instruction to pour water in the cup at four random time instants during each interaction tasks (40 total pour instructions). The robot makes a decision whether to execute the pouring task or not, based on the anticipated activity and object affordance. The robot considers the three top scored anticipations for taking the decision following the *robot anticipation metric*.

We obtain a success rate of 85%, which is the fraction of times the robot correctly identifies its response ('pour' or 'not pour'). Out of the 6 failed instances, 3 instances are false-negatives, i.e., the robot anticipated an interaction with the cup when no interaction occurred in future. Videos showing the results of our robotic experiments and code are available at: `http://pr.cs.cornell.edu/anticipation/`.

9 Conclusion

In this work, we considered the problem of detecting the past human activities as well as anticipating the future using object affordances. We showed how the anticipation of future activities can be used by a robot to perform look-ahead planning of its reactive responses. We modeled the human activities and object affordances in the past using a rich graphical model (CRF), and extended it to include future possible scenarios. Each possibility was represented as a potential graph structure and labeling over the graph (which includes discrete labels as well as human and object trajectories), which we called ATCRF. We used importance sampling techniques for estimating and evaluating the most likely future scenarios. The structure of the ATCRF was obtained by first considering the potential graph structures that are close to the ground-truth ones by approximating the graph with only additive features. We then designed moves to explore the space of likely graph structures. We showed that anticipation can improve performance of detection of even past activities and affordances. We also extensively evaluated our algorithm, against baselines, on the tasks of anticipating activity and affordance labels as well as the object trajectories.

In the recently growing field of RGB-D vision, our work thus shows a considerable advance by improving the state-of-the-art results on both the detection and anticipation tasks. We have focused on the algorithms for estimating the graph structure for both past and future activities while using given noisy skeleton and object tracks. Improvements to object perception would further improve these results. In our experiments, we see that there is still a large gap between the detection performance with and without ground-truth temporal segmentation. Incorporating additional priors about the activities in future work would improve the estimation of the graph structure. Also, we see that the anticipation accuracies fall rapidly with future prediction time. We believe that modeling larger temporal-range dependencies and hierarchical structure of activities is an interesting direction to explore for obtaining better anticipation.

Acknowledgments

The authors would like to thank Rudhir Gupta and Jerry Yeh for useful discussions. This work was partially supported by ARO award W911NF-12-1-0267, and by Microsoft Faculty Fellowship and NSF Career Award to one of us (Saxena).

References

[1] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *CVPR*, 2012.

[2] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *CVPR*, 2012.

[3] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *CVPR*, 2012.

[4] J.-K. Min and S.-B. Cho, "Activity recognition based on wearable sensors using selection/fusion hybrid ensemble," in *SMC*, 2011.

[5] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *IJRR*, 2013.

[6] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from rgbd images," in *ICRA*, 2012.

[7] B. Ni, G. Wang, and P. Moulin, "Rgbd-hudaact: A color-depth video database for human daily activity recognition," in *ICCV Workshop on CDC4CV*, 2011.

[8] E. Guizzo and E. Ackerman, "The rise of the robot worker," *Spectrum, IEEE*, vol. 49, no. 10, pp. 34 –41, October 2012.

[9] S. Nikolaidis and J. Shah, "Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy," in *HRI*, 2013.

[10] J. Gibson, *The Ecological Approach to Visual Perception.* Houghton Mifflin, 1979.

[11] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *CVPR*, 2011.

[12] Z. Xing, J. Pei, G. Dong, and P. S. Yu, "Mining Sequence Classifiers for Early Prediction," in *SIAM ICDM*, 2008.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[14] J. Niebles, C. Chen, and L. Fei-fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *ECCV*, 2010.

[15] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *CVPR*, 2011.

[16] B. Laxton, L. Jongwoo, and D. Kriegman, "Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video," in *CVPR*, 2007.

[17] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *CVPR*, 2009.

[18] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *CVPR*, 2011.

[19] S. Sadanand and J. Corso, "Action bank: A high-level representation of activity in video," in *CVPR*, 2012.

[20] M. Rohrbach, M. Regneri, M. A., S. Amin, M. Pinkal, and B. Schiele, "Script data for attribute-based recognition of composite activities," in *ECCV*, 2012.

[21] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Attribute learning for understanding unstructured social activity," in *ECCV*, 2012.

[22] M. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *ICCV*, 2011.

[23] M. Hoai and F. De la Torre, "Max-margin early event detectors," in *CVPR*, 2012.

[24] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comp Surveys (CSUR)*, 2011.

[25] W. Yang, Y. Wang, and G. Mori, "Recognizing human actions from still images with latent poses," in *CVPR*, 2010.

[26] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: Using spatial and functional compatibility for recognition," *IEEE PAMI*, vol. 31, no. 10, pp. 1775–1789, 2009.

[27] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *CVPR*, 2010.

[28] Y. Jiang, M. Lim, and A. Saxena, "Learning object arrangements in 3d scenes using human context," in *ICML*, 2012.

[29] S. Tran and L. S. Davis, "Event modeling and recognition using markov logic networks," in *ECCV*, 2008.

[30] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *NIPS*, 2011.

[31] Z. Jia, A. Gallagher, A. Saxena, and T. Chen, "3d-based reasoning with blocks, support, and stability," in *CVPR*, 2013.

[32] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE TPAMI*, 2012.

[33] D. L. Ly, A. Saxena, and H. Lipson, "Co-evolutionary predictors for kinematic pose inference from rgbd images," in *GECCO*, 2012.

[34] H. Zhang and L. E. Parker, "4-dimensional local spatio-temporal features for human activity recognition," in *IROS*, 2011.

[35] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video," in *ICCV*, 2009.

[36] S. Satkin and M. Hebert, "Modeling the temporal extent of actions," in *ECCV*, 2010.

[37] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *ICCV*, October 2007.

[38] M. H. Nguyen, L. Torresani, F. De la Torre, and C. Rother, "Weakly supervised discriminative localization and classification: a joint learning process," in *ICCV*, 2009.

[39] T. Simon, M. H. Nguyen, F. De la Torre, and J. F. Cohn, "Action unit detection with segment-based svms," in *CVPR*, 2010.

[40] X. Xuan and K. Murphy, "Modeling changing dependency structure in multivariate time series," in *ICML*, 2007.

[41] Z. Harchaoui, F. Bach, and E. Moulines, "Kernel change-point analysis," in *NIPS*, 2008.

[42] S. Oh, J. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems," *IJCV*, 2008.

[43] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Nonparametric Bayesian learning of switching linear dynamical systems," in *NIPS 21*, 2009.

[44] M. Hoai, Z. Lan, and F. De la Torre, "Joint segmentation and classification of human actions in video," in *CVPR*, 2011.

[45] M. Hoai and F. De la Torre, "Maximum margin temporal clustering," in *AISTATS*, 2012.

[46] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware modeling and recognition of activities in video," in *CVPR*, 2013.

[47] Z. Wang, Q. Shi, C. Shen, and A. van den Hengel, "Bilinear programming for human activity recognition with unknown mrf graphs," in *CVPR*, 2013.

[48] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *IROS*, 2009.

[49] K. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *ECCV*, 2012.

[50] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *RSS*, 2012.

[51] Z. Wang, M. Deisenroth, H. B. Amor, D. Vogt, B. Scholkopf, and J. Peters, "Probabilistic modeling of human movements for intention inference," in *RSS*, 2012.

[52] A. Dragan and S. Srinivasa, "Formalizing assistive teleoperation," in *RSS*, 2012.

[53] S. Hongeng and R. Nevatia, "Large-scale event detection using semi-hidden markov models," in *ICCV*, 2003.

[54] P. Natarajan and R. Nevatia, "Coupled hidden semi markov models for activity recognition," in *WMVC*, 2007.

[55] S. Gong and T. Xiang, "Recognition of group activities using dynamic probabilistic networks," in *ICCV*, 2003.

[56] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden-state conditional random fields," *IEEE PAMI*, 2007.

[57] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition," in *ICCV*, 2005.

[58] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *NIPS*, 2004.

[59] Q. Shi, L. Wang, L. Cheng, and A. Smola, "Human action segmentation and recognition using discriminative semi-markov models," *IJCV*, 2011.

[60] Z. Khan, T. Balch, and F. Dellaert, "Mcmc-based particle filtering for tracking a variable number of interacting targets," *IEEE PAMI*, 2005.

[61] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," in *CVPR*, 2009.

[62] D. Fox, "Kld-sampling: Adaptive particle filters," in *NIPS*, 2001.

[63] P. Jensfelt, D. Austin, O. Wijk, and M. Andersson, "Feature based condensation for mobile robot localization," in *ICRA*, 2000.

[64] D. Schulz, D. Fox, and J. Hightower, "People tracking with anonymous and id-sensors using rao-blackwellised particle filters," in *IJCAI*, 2003.

[65] A. Doucet, N. d. Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *UAI*, 2000.

[66] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. SP*, 2002.

[67] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *AAAI*, 2002.

[68] "Openni," https://github.com/OpenNI/OpenNI.

[69] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *ICRA*, 2002.

[70] H. Kjellström, J. Romero, and D. Kragic, "Visual object-action recognition: Inferring object affordances from human demonstration," *CVIU*, vol. 115, no. 1, pp. 81–90, 2011.

[71] J. Sun, J. L. Moore, A. Bobick, and J. M. Rehg, "Learning visual object categories for robot affordance prediction," *IJRR*, 2009.

[72] T. Hermans, J. M. Rehg, and A. Bobick, "Affordance prediction via learned object attributes," in *ICRA: Workshop on Semantic Perception, Mapping, and Exploration*, 2011.

[73] H. Grabner, J. Gall, and L. Van Gool, "What makes a chair a chair?" in *CVPR*, 2011.

[74] A. Gupta, S. Satkin, A. Efros, and M. Hebert, "From 3d scene geometry to human workspace," in *CVPR*, 2011.

[75] V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. Efros, "Scene semantics from long-term observation of people," in *ECCV*, 2012.

[76] Y. Jiang, H. S. Koppula, and A. Saxena, "Hallucinated humans as the hidden context for labeling 3d scenes," in *CVPR*, 2013.

[77] J. J. Faraway, M. P. Reed, and J. Wang, "Modelling three-dimensional trajectories by using bezier curves with application to hand motion," *JRSS Series C*, vol. 56, pp. 571–585, 2007.

[78] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," in *NIPS*, 2011.

[79] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," in *CVPR*, 2007.

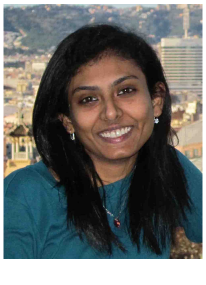
[80] P. Hammer, P. Hansen, and B. Simeone, "Roof duality, complementation and persistency in quadratic 0–1 optimization," *Mathematical Prog.*, vol. 28, no. 2, pp. 121–155, 1984.

[81] P. F. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, 2004.

[82] T. Joachims, T. Finley, and C. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, 2009.

[83] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, CMU, Robotics Institute, August 2010.

Ashutosh Saxena is in the Faculty of the Computer Science department at Cornell University. His research interests include machine learning, robotics and computer vision. He received his Ph.D. in 2009 from Stanford University, and his B.Tech. in 2004 from IIT Kanpur, India. He has won best paper awards in 3DRR, RSS and IEEE ACE. He was named a co-chair of IEEE technical committee on robot learning. He has also received Sloan Fellowship in 2012, NSF Career award in 2013, and RSS Early Career Award in 2014. He has developed robots that perform household chores such as unload items from a dishwasher, arrange a disorganized house, checkout groceries, etc. Previously, he has developed Make3D, an algorithm that converts a single photograph into a 3D model.



Hema S. Koppula is a PhD student in the Computer Science department at Cornell University. Her research lies at the intersection of machine learning, computer vision and robotics: she is interested in understanding people from visual data to build smart assistive devices. She has developed machine learning algorithms for perceiving environments from RGB-D sensors such as scene understanding, activity detection and anticipation. She has won the best student paper award at RSS and is a Google PhD Fellow.