First-Person Activity Forecasting with Online Inverse Reinforcement Learning

Nicholas Rhinehart and Kris M. Kitani
Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213
{nrhineha,kkitani}@cs.cmu.edu

Abstract

We address the problem of incrementally modeling and forecasting long-term goals of a first-person camera wearer: what the user will do, where they will go, and what goal they seek. In contrast to prior work in trajectory forecasting, our algorithm, DARKO*, goes further to reason about semantic states (will I pick up an object?), and future goal states that are far in terms of both space and time.* DARKO *learns and forecasts from first-person visual observations of the user's daily behaviors via an Online Inverse Reinforcement Learning (IRL) approach. Classical IRL discovers only the rewards in a batch setting, whereas* DARKO *discovers the states, transitions, rewards, and goals of a user from streaming data. Among other results, we show* DARKO *forecasts goals better than competing methods in both noisy and ideal settings, and our approach is theoretically and empirically no-regret.*

1. Introduction

Our long-term aim is to develop an AI system that can learn about a person's intent and goals by continuously observing their behavior. In this work, we progress towards this aim by proposing an online Inverse Reinforcement Learning (IRL) technique to learn a decision-theoretic human activity model from video captured by a wearable camera. The use of a wearable camera is critical to our task, as human activities must be observed up close and across large environments. Imagine a person's daily activities—perhaps they are at home today, moving about, completing tasks. Perhaps they are a scientist that conducts a long series of experiments across various stations in a laboratory, or they work in an office building where they walk about their floor, get coffee, *etc*. As people tend to be very mobile, a wearable camera is ideal for observing a person's behavior.

Since our task is to continuously learn human behavior models (*i.e*., a policy) from observed behavior captured with a wearable camera, our task is best described as an online IRL problem. The problem is an *inverse* Reinforcement Learning problem because the underlying reward or cost function of the person is unknown. We must infer it along with the policy from the demonstrated behaviors. Our task is also an *online learning problem*, because our algorithm must continuously learn as a part of a life-long process. From this perspective, we must develop an online learning approach that learns effectively over time.

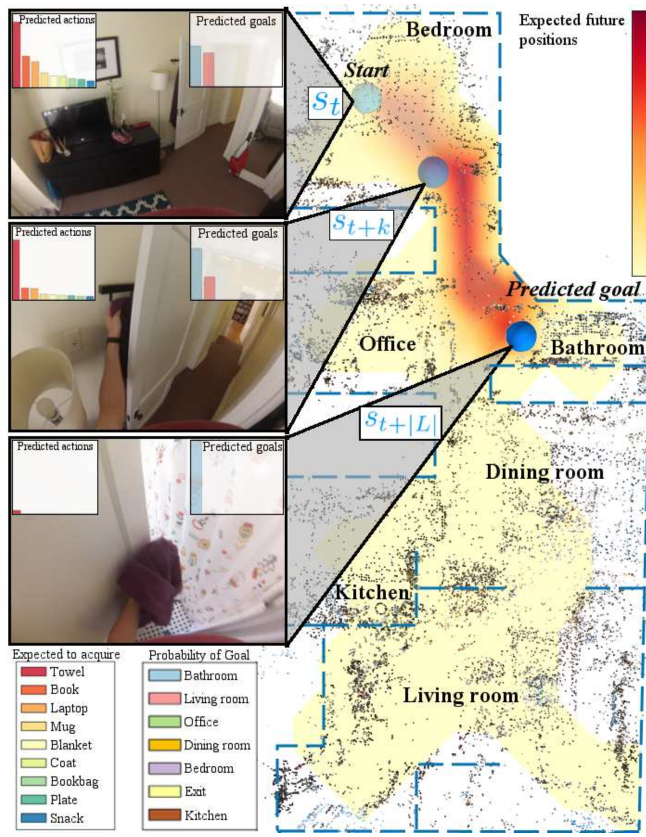

Figure 1: **Forecasting future behavior from first-person video.** Overhead map shows likely future goal states. s_i is user *state* at time i. Histogram insets display predictions of user's long-term semantic goal (inner right) and acquired objects (inner left).

(a) Lab environment

(b) Home 1 environment

Figure 2: Sparse SLAM points (2a) and offline dense reconstruction (2b) using [5] for two of our dataset environments.

We present an algorithm that *incrementally learns spatial and semantic intentions* (where you will go and what you will do) of a first-person camera wearer. By tracking the goals a person achieves, the algorithm builds a set of possible futures. At any time, the user's future is predicted among this set of goals. We term our algorithm "Discovering Agent Rewards for K-futures Online" (DARKO), as it learns to associate rewards with semantic states and actions from demonstrations to predict among K possible goals.

Contributions: To the best of our knowledge, we present the first application of ideas from online learning theory and inverse reinforcement learning to the task of continuously learning human behavior models with a wearable camera. Our proposed algorithm is distinct from traditional IRL problems as we jointly discover states, transitions, goals, and the reward function of the underlying Markov Decision Process model. Our proposed human behavior model also goes beyond first-person trajectory forecasting and allows for the prediction of future human activities that can happen outside the immediate field of view and far into the future.

2. Related Work

First-person vision (FPV): Wearable cameras have been used for various human behavior understanding tasks [4, 10, 12, 16, 22] because they give direct access to detailed visual information about a person's actions. Leveraging this feature of FPV, recent work has shown that it is possible to predict where people will look during actions [12] and how people will use the environment [20].

Decision-Theoretic Modeling: Given agent demonstrations, the task of *inverse reinforcement learning* (IRL) is to recover a *reward function* of an underlying Markov Decision Process (MDP) [1]. IRL has been used to model taxi driver behavior [34] and pedestrian behavior [35, 7]. In contrast to previous work, we go beyond physical trajectory forecasting by reasoning over future object interactions and predicting future goals in terms of scene types.

Online Learning Theory: The theory of learning to making optimal predictions from streaming data is well studied [23] but is rarely used in computer vision, compared to the more prevalent application of supervised learning theory. We believe, however, that the utility of online learning theory is likely to increase as the amount of available data for processing is ever increasing. While the concept of online learning has been applied to inverse reinforcement learning [18], the work was primarily theoretic in nature and has found limited application.

Trajectory forecasting: Physical trajectory forecasting has received much attention from the vision community. Multiple human trajectory forecasting from a surveillance camera was investigated by [14]. Other trajectory forecasting approaches use demonstrations observed from a bird's-eye view; [31] infers latent goal locations and [2] employ LSTMs to jointly reason about trajectories of multiple humans. In [25], the model forecasted short-term future trajectories of a first-person camera wearer by retrieving the nearest neighbors from a dataset of first-person trajectories under an obstacle-avoidance cost function, with each trajectory representing predictions of where the user will move in view of the frame; in [26], a similar model with learned cost function is extended to multiple users.

Predicting Future Behavior: In [6, 21], the tasks are to recognize an unfinished event or activity. In [6], the model predicts the onset for a single facial action primitive, *e.g.* the completion of a smile, which may take less than a second. Similarly, [21] predicts the completion of short human to human interactions. In [9], a hierarchical structured SVM is employed to forecast actions about a second in the future, and [28] demonstrates a semi-supervised approach for forecasting human actions a second into the future. Other works predict actions several seconds into the future [3, 8, 11, 29]. In contrast, we focus on high-level transitions over a sequence of future actions that may occur outside the frame of view, and take a longer time to complete (in our dataset, the mean time to completion is 21.4 seconds).

3. Online IRL with DARKO

Our goal is to forecast the future behaviors of a person from a continuous stream of video captured by a wearable camera. Given a continuous stream of FPV video, our approach extracts a sequence of state variables $\{s_1, s_2, \dots\}$ using a portfolio of visual sensing algorithms (*e.g.*, SLAM, stop detection, scene classification, action and object recognition). In an online fashion, we segment this state sequence into episodes (short trajectories) by discovering terminal goal states (*e.g.*, when a person stops). Using the most recent episode, we adaptively solve the inverse reinforcement learning problem using online updates. Solving the IRL problem in an online fashion means that we incrementally learn the underlying decision process model.

3.1. First-Person Behavior Model

A Markov Decision Process (MDP) is commonly used to model the sequential decision process of a rational agent. In our case, we use it to describe the activity of a person with a wearable camera. In a typical reinforcement learning problem, all elements of the MDP are assumed to be known and the task is to estimate an optimal policy $\pi(a|s)$, that maps a state s to an action a, by observing rewards. In our novel online inverse formulation, every element of the MDP, including the policy, is unknown and must be inferred as new video data arrives. Formally, an MDP is defined as:

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, T(\cdot,\cdot), R_\theta(\cdot,\cdot)\}.$$

States: $\mathcal{S}$ is the state space: the set of states an agent can visit. In our online formulation, $\mathcal{S}$ is initially empty, and must be expanded as new states are discovered. We define a state s as a vector that includes the location of the person (3D position), the last place the person stopped (a previous goal state), and information about any object that the person might be holding. Formally, a state $s \in \mathcal{S}$ is denoted as:

$$s = [x, y, z, o_1 \dots, o_{|\mathcal{O}|}, h_1, \dots h_{|\mathcal{K}|}].$$

The triplet $[x, y, z]$ is a discrete 3D position. To obtain the position, we use a monocular visual SLAM algorithm [15] to localize the agent in a continuously built map.

The vector $o_1 \dots, o_{|\mathcal{O}|}$ encodes any objects that the person is currently holding. We include this information in the state vector because the objects a user acquires are strongly correlated to the intended activity [4]. $o_j = 1$ if the user has object j in their possession and zero otherwise. $\mathcal{O}$ is a set of pre-defined objects available to the user. $\mathcal{K}$ is a set of pre-defined scene types available to the user, which can be larger than the true number of scene types. The vector $h_1, \dots h_K$ encodes the last scene type the person stopped. Example scene types are `kitchen` and `office`. $h_i = 1$ if the user last arrived at scene type i and is zero otherwise.

Goals: We also define a special type of state called a *goal state* $s \subset \mathcal{S}_g$, to denote states where the person has achieved a goal. We assume that when a person stops, their location in the environment is a goal. We detect goal states by using a velocity-based stop detector. Whenever a goal state is encountered, the sequence of states since the last goal state to the current goal state is considered a completed episode ξ. The set of goals states $\mathcal{S}_g \subset \mathcal{S}$ expands with each detection. We explain later how $\mathcal{S}_g$ is used to perform goal forecasting.

Actions: $\mathcal{A}$ is the set of actions. $\mathcal{A}$ can be decomposed into two parts: $\mathcal{A} = \mathcal{A}_m \cup \mathcal{A}_c$. The act of moving from one location in the environment to another location is denoted as $a_m \in \mathcal{A}_m$. Like $\mathcal{S}$, $\mathcal{A}_m$ must be built incrementally. The set $\mathcal{A}_c$ is the set of possible `acquire` and `release` actions of each object: $\mathcal{A}_c = \{\texttt{acquire}, \texttt{release}\} \times \mathcal{O}$. The act of releasing or picking up an object is denoted as $a_c \in \mathcal{A}_c$. Each action a_c must be detected. We do so with an image-based first-person action classifier. More complex approaches could improve performance [13].

Transition Function: The transition function $T : (s, a) \mapsto s'$ represents how actions move a person from one state to the next state. T is constructed incrementally as new states are observed and new actions are performed. In our work, T is built by keeping a table of observed (s, a, s') triplets, which describes the connectivity graph over the state space. More advanced methods could also be used to infer more complex transition dynamics [27, 30].

Reward Function: $R(s, a; \theta)$ is an instantaneous reward function of action a at state s. We model R as the inner product between a vector of features $f(s, a)$ and a vector of weights θ. The reward function is essential in value-based reinforcement learning methods (in contrast to policy search methods) as it is used to compute the policy $\pi(a|s)$. In the maximum entropy setting, the policy is given by $\pi(a|s) \propto e^{Q(s,a)-V(s)}$, where the value functions $V(s)$ and $Q(s, a)$ are computed from the reward function by solving the Bellman equations [34]. In our context, we learn the reward function online.

Intuitively, we would like the features f of the reward function to incorporate information such as the position in an environment or objects in possession, since it is reasonable to believe that the goal of many activities is to reach a certain room or to retrieve a certain object. To this end, we define the features of the reward to mirror the information already contained in the state s_t: the position, previous scene type, and objects held. To be concrete, the feature vector $f(s, a)$ is the concatenation of the 3-d position coordinates $[x, y, z]$, a K-dimensional indicator vector over previous goal state type and a $|\mathcal{O}|$-dimensional indicator vector over held objects. We also concatenate a $|\mathcal{A}_c|$-dimensional indicator vector over actions $a_c \in \mathcal{A}_c$.

3.2. The DARKO Algorithm

We now describe our proposed algorithm for incrementally learning all MDP parameters, most importantly the reward function, given a continuous stream of first-person video (see DARKO in Algorithm 1). The procedure begins by initializing s, reward parameters θ, empty state space $\mathcal{S}$, goal space $\mathcal{S}_g$, transition function T, and current episode ξ.

State Space Update: Image frames are obtained from a first-person camera (the NEWFRAME function), and SLAM is used to track the user's location (lines 4 and 5). An image-based action detection algorithm, ACTDET, detects hand-object interactions a_c and decides movements a_m as a function of current and previous position. While we provide an effective method for ACTDET, our focus is to integrate (rather than optimize) its outputs. Lines 7 and 8 show how the trajectory is updated and MDP parameters of state space and transition function are expanded. Line 9 represents a collection of generalized forecasting tasks (see Section 4.4), such as the computation of future goal posterior and trajectory forecasting.

Goal Detection: In order to discover goals, we use a stop-detection algorithm GOALDET, by using the camera velocity computed from SLAM (Line 10). If a goal state has been detected, that terminal state is added to the set of goal states $\mathcal{S}_g$. The detection of a terminal state also marks the end of an episode ξ. The previous goal state type is also updated for the next episode. Again, while we provide an effective method for GOALDET, our focus is to integrate (rather than optimize) its outputs.

Online IRL: With the termination of each episode ξ, the reward function R and corresponding policy π are updated via the reward parameters θ (Line 13). The parameter update uses a sequence of demonstrated behavior via the episode ξ, and the current parameters of the MDP. More specifically, ONLINEIRL (Algorithm 2) performs online gradient descent on the likelihood under the maximum entropy distribution by updating current parameters of the reward function. The gradient of the loss can be shown to be the difference between expected feature counts $\bar{f}$ and empirical feature counts $\hat{f}$ of the current episode. Computing the gradient requires solving the soft value iteration algorithm of [33]. We include a projection step to ensure $\|\theta\|_2 \leq B$.

To the best of our knowledge, this is the first work to propose an online algorithm for maximum entropy IRL in the streaming data setting. Following the standard procedure for ensuring good performance of an online algorithm, we analyze our algorithm in terms of the regret bound. The regret $\mathcal{R}_t$ of any online algorithm is defined as:

$$\mathcal{R}_t = \sum_{i=0}^{t} l_t(\xi_t; \theta_t) - \min_{\theta^*} \sum_{i=0}^{t} l(\xi_t; \theta^*).$$

The regret is the cumulative difference between the performance of the online model using current parameter θ versus the best hindsight model using the best parameters θ^*. The loss l_t is a function of the t'th demonstrated trajectory, and measures how well the model explains the trajectory.

In our setup, the loss function is defined as $l_t(\xi_t; \theta) = -\frac{1}{|\xi_t|} \sum_{i=0}^{|\xi_t|} \log \pi_\theta(a_i|s_i)$. It can be shown[1] that the regret

[1]Proof of the regret bound is given in the Appendix

Algorithm 1 DARKO: Discovering Agent Rewards for K-futures Online

1: **procedure** DARKO(SLAM, ACTDET, GOALDET)
2: $s \leftarrow \mathbf{0}, \theta = \mathbf{0}, \mathcal{S} = \{\}, \mathcal{S}_g = \{\}$, T.INIT(), $\xi = []$
3: **while** True **do**
4: frame $\leftarrow$ NEWFRAME()
5: $[x, y, z] \leftarrow$ SLAM.TRACK(frame)
6: $a \leftarrow$ ACTDET($[x, y, z]$, frame)
7: $\xi \leftarrow \xi \oplus (s, a), \mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$
8: T.EXPAND(s, a), $s \leftarrow T(s, a)$
9: ▶ Goal forecasting, trajectory forecasting, ...
10: is_goal $\leftarrow$ GOALDET(s, frame, $\mathcal{S}_g$)
11: **if** is_goal **then**
12: $\mathcal{S}_g \leftarrow \mathcal{S}_g \cup \{s\}$
13: $\pi, \theta \leftarrow$ ONLINEIRL($\theta, \mathcal{S}, T, \xi, \mathcal{S}_g$)
14: $s \leftarrow T(s, a = \text{at\_goal}), \xi = []$
15: **end if**
16: **end while**
17: **end procedure**

Algorithm 2 Online Inverse Reinforcement Learning

procedure ONLINEIRL($\theta, \mathcal{S}, T, \xi, \mathcal{S}_g; \lambda, B$)
$\overline{f}_i = \sum_{(s,a)\in\xi} f(s, a)$
▶ Compute $R(s, a; \theta)\ \forall s \in \mathcal{S}, a \in \mathcal{A}$
$\pi \leftarrow$ SOFTVALUEITERATION($R, \mathcal{S}, \mathcal{S}_g, T$)
$\hat{f}_i \leftarrow E_\pi[f(s, a)]$
$\theta \leftarrow \text{proj}_{\|\theta\|_2 \leq B}(\theta - \lambda(\overline{f}_i - \hat{f}_i))$
return π, θ
end procedure

of our online algorithm is bounded as follows:

$$\mathcal{R}_t \leq 2B\sqrt{2td}, \tag{1}$$

where B is a norm bound on θ, d is feature dimensionality, and t is the episode. The average regret $\frac{\mathcal{R}_t}{t}$ approaches zero as t grows since the bound is sub-linear in t. Verification of this no-regret property is shown in the experiments.

4. Generalized Activity Forecasting

Without Line 9, Algorithm 1 only describes our online IRL process to infer the reward function. In order to make incremental predictions about the person's future behaviors online, we can leverage the current MDP and reward function. An important function which lays the basis for predicting future behaviors is the state visitation function, denoted D. We now show how D can be modified to perform generalized queries about future behavior.

4.1. State Visitation Function D

Using the current estimate of the MDP and the reward function, we can compute the policy of the agent. Using the policy, we can forward simulate a distribution of all possible futures. This distribution is called the state visitation distribution [33]. More formally, the posterior expected count of future visitation to a state s_x can be defined as

$$D_{s_x|\xi_{0\to t}} \triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau = s_x)\right]. \quad (2)$$

This quantity represents the agent's expectation to visit each state in the future given the partial trajectory. $\xi_{0\to t}$ indicates a partial trajectory starting at time 0 and ending at time t. The expectation is taken under the maximum causal entropy distribution, $P(\xi_{t+1\to T}|\xi_{0\to t})$, which gives the probability of a future trajectory given the current trajectory. I is the indicator function, which counts agent visits to s_x. Equation 2 is estimated by sampling trajectories from $\pi_\theta(a|s)$.

4.2. Activity Forecasting with State Subsets

In this work, we extend the idea of state visitations to a single state s_x to a more general *subset of states* $\mathcal{S}_p$. While a generalized prediction task was not particularly meaningful in the context of trajectory prediction [7, 34], predictions over a subset of states now represents semantically meaningful concepts in our proposed MDP. By using the state space representation of our first-person behavior model, we can construct subsets of the state space that have interesting semantic meaning, such as "having an object o_i" or "all states closest to goal k with $\mathcal{O}_j$ set of objects."

Formally, we define the expected count of visitation to a subset of states $\mathcal{S}_p$ satisfying some property p:

$$D_{\mathcal{S}_p|\xi_{0\to t}} \triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau \in \mathcal{S}_p)\right] \quad (3)$$

$$= \sum_{s_x\in\mathcal{S}_p} \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau = s_x)\right]$$

$$= \sum_{s_x\in\mathcal{S}_p} D_{s_x|\xi_{0\to t}}. \quad (4)$$

Equation 4 is essentially marginalizing over the state subspace of Equation 2. Derivation of two other inference tasks is given in the Appendix: (1) expected visitation prediction of performing an action after arriving at a subspace, (2) expected joint action-state subspace visitation prediction.

4.3. Forecasting Trajectory Length

In the following, we present a method to predict the length of the future trajectory. Formally, we can denote the expected trajectory length:

$$\hat{\tau}_{\xi_{t+1\to T}|\xi_{0\to t}} \triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\,|\xi_{t+1\to T}| \quad (5)$$

Consider evaluating $D_{\mathcal{S}_p|\xi_{0\to t}}$ from Equation 4 by setting $\mathcal{S}_p = \mathcal{S}$, that is, by considering the expected future visitation count to the entire state space. Then,

$$\begin{aligned} D_{\mathcal{S}|\xi_{0\to t}} &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau \in \mathcal{S})\right] \\ &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} 1\right] \\ &= \mathbb{E}\,|\xi_{t+1\to T}| = \hat{\tau}_{\xi_{t+1\to T}|\xi_{0\to t}} \end{aligned} \quad (6)$$

where $|\xi|$ indicates the number of states in trajectory ξ.

4.4. Future Goal Prediction

As previously described, we wish to predict the final goal of a person's action sequence. For example, if I went to the study to pick up a cup, how likely am I to go to the kitchen versus the living room? This problem can be posed as solving for the MAP estimate of $P(g|\xi)\forall g \in \mathcal{S}_g$, the posterior over goals. It describes *what goal the user seeks given their current trajectory*, defined as:

$$P(g|\xi_{0\to t}) \propto P(g)e^{V_{s_t}(g)-V_{s_0}(g)}, \quad (7)$$

where $V_{s_i}(g)$ is the value of g with respect to a partial trajectory that ends in s_i. Notice that the likelihood term is exponentially proportional to the value difference between the start state s_0 and the current state s_t. In this way, the likelihood encodes the progress made towards a goal g in terms of the value function.

5. Experiments

We first present the dataset we collected. Then, we discuss our methods for goal discovery and action recognition. To reiterate, our focus is not to engineer these methods, but instead to make intelligent use of their outputs in DARKO for the purpose of behavior modeling. We compare DARKO's performance versus several baselines on the task of goal forecasting, and show DARKO's performance is superior. Then, we analyze DARKO's performance under less noisy conditions, to illustrate how it improves when provided with more robust goal discovery and action detection algorithms. Next, we illustrate DARKO 's empirical no-regret performance, which further shows it is an efficient online learning algorithm. Finally, we present trajectory length forecasting results, and find that our length forecasts exhibit low median error. Additional analyses including feature ablation and incorporating uncertainty from goal discovery are presented in the Appendix.

5.1. First-Person Continuous Activity Dataset

We collected a dataset of sequential goal-seeking behavior in several different environments such as home, office and laboratory. The users recorded a series of activities that naturally occur in each scenario. Each user helped design the script they followed, which involved their prior assumptions about what objects they will use and what goal they will seek. An example direction from a script is "obtain a snack and plate in kitchen, eat at dining room table."

Users wore a hat-mounted Go-Pro Hero camera with 94° vertical, 123° horizontal FOVs. Our dataset is comprised of 5 user environments, and includes over 250 actions with 19 objects, 17 different scene types, at least 6 activity goals per environment, and about 200 high-level activities (trajectories). In each environment, the user recorded 3–4 long sequences of high-level activities, where each sequence represents a full day of behavior. Our dataset represents over 15 days of recording.

For evaluation, all ground truth labels of objects (*e.g.* cup, backpack), actions (*i.e.* acquire, release) and goals (*e.g.* kitchen, bedroom) were first manually annotated. A goal label correspond to *when* a high-level direction was completed, and *in which scene* it was completed, *e.g.* (dining room, time=65s). An action label indicates when an activity was performed, *e.g.* (acquire, cup, time=25s). Further details are presented in the Appendix.

5.2. Goal Discovery and Action Recognition

We describe two goal discovery methods and an action recognition method that can serve as input to DARKO. With respect to Algorithm 1, these are GOALDET and ACTDET.

Scene-based Goal Discovery: This model assumes that if a scene classifier is very confident in the scene type for several images frames, the camera wearer must be in a meaningful place in the environment (*i.e.*, kitchen, bedroom, office). We use the output of a scene classifier from [32] (GoogLeNet model) on every frame from the wearable camera. If the mean scene classifier probability for a scene type is above a threshold ρ_g for 20 consecutive image frames, then we add the current state s_t to the set of goals $\mathcal{S}_g$.

Stop-based Goal Discovery: This model assumes that when a person stops, they are at an important location. Using SLAM's 3D camera positions, we apply a threshold on velocity to detect stops in motion. When a stop is detected, we add the current state s_t to the set of goals $\mathcal{S}_g$. In Table 1, temporal accuracies are computed by counting detections within 3-second windows of ground truth labels as true positives; for the scene-based method, a true positive also requires the scene type to match the ground truth scene type. Stop-based discovery is reliable across all environments, thus, we use it as our primary goal discovery method.

Image-based Object Recognition: We designed an object recognition approach that classifies the object the user interacts with at every temporally-labeled window. It overwrites the ground-truth object label with its detection. The approach first detects regions of `person` in each frame with [19] to focus on objects near the visible hands, which are cropped with context and fed into an image-classifier trained on ImageNet [24]. The outputs are remapped to our object set, and a final classification is produced by taking the maximum across objects. The per-action classification accuracies in Table 1 demonstrate the method can produce reasonable action classifications across all scenes. While imperfect, these detections serve as useful input to DARKO.

Figure 3: **Goal forecasting examples:** A temporal sequence of goal forecasting results is shown in each row from left to right, with the forecasted goal icons and sorted goal probabilities inset (green: $P(g^*|\xi)$, red: $P(g_i \neq g^*|\xi)$). *Top*: the scientist acquires a sample to boil in the gel electrophoresis room. *Middle*: the user gets a textbook and goes to the lounge. *Bottom*: the user leaves their apartment.

Method	Home 1	Home 2	Office 1	Office 2	Lab 1
Scene Discovery	0.93	0.24	0.62	0.49	0.32
Stop Discovery	0.62	0.68	0.67	0.69	0.73
Act. Recognition	0.64	0.63	0.66	0.56	0.71

Table 1: **Goal Discovery and Action Recognition.** The per-scene goal discovery and action recognition accuracies are shown for our methods. A 3-second window is used around every goal discovery to compute accuracy.

Method	Home 1	Home 2	Office 1	Office 2	Lab 1
DARKO	**0.524**	**0.378**	**0.667**	**0.392**	**0.473**
MMED [6]	0.403	0.299	0.600	0.382	0.297
RNN	0.291	0.274	0.397	0.313	0.455
Logistic	0.458	0.297	0.569	0.323	0.348
Uniform	0.181	0.098	0.233	0.111	0.113

Table 2: **Goal Forecasting Results (Visual Detections):** Proposed goal posterior (Sec.4.4) achieves best $\overline{P}_{g^*}$ (mean probability of true goal).

5.3. Goal Forecasting Performance

At every time step, our method predicts the user's goal or final destination (*e.g.*, bedroom, exit) as described in Section 4.4 and shown in Figure 3. To understand the goal prediction reliability, we compare our approach to several baseline methods for estimating the goal posterior $P(g|\xi_{0\to t})$, where g is a goal and $\xi_{0\to t}$ is the observed state sequence up to the current time step. Each baseline requires the state tracking and goal discovery components of DARKO.

Uniform Model (Uniform): This model returns a uniform posterior over possible goals $P_n(g) = 1/K_n$ known at the current episode n, defining worst case performance.

Logistic Regression Model (Logistic): A logistic regression model $P_n(g|s_t)$ is fit to map states s_t to goals g.

Max-Margin Event Detection (MMED) [6]: A set of max-margin models $P_n(g|\phi(s_{t:t-w}))$ are trained to map features ϕ of a w-step history of state vectors $s_{t:t-w}$ to a goal score. We use the best performing `sumL1norm` features provided with the publicly available code.

RNN Classifier (RNN): An RNN is trained to predict $P_n(g|\xi_{0\to t})$. We experimented with a variety of parameters (see the Appendix) and report the best results.

Since all methods above are online algorithms, each of the models P_n is updated after every episode n. In order quantify performance with a single score, we use the mean probability assigned to the ground truth goal type g^* over all episodes $\{\xi_n\}_{n=1}^N$ as $\overline{P}(g^*|\{\xi_n\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^N\sum_{t=1}^{T_n} P_n(g|\xi_{nt})$ (also denoted $\overline{P}_{g^*}$). The goal forecasting performance results are summarized in Table 2 using the above metric.

5.4. Goal Forecasting with Perfect Visual Detectors

The experimental results up to this point have exclusively used visual detectors as input (*e.g.*, SLAM, scene classification, object recognition). While we have shown that our approach learns meaningful human activity models from real computer vision input, we would also like to understand how our online IRL method performs when decoupled from the noise of the vision-based input. We perform the same experiments described in Section 5.3 but with idealized (ground truth) inputs for goal discovery and action

Method	Home 1	Home 2	Office 1	Office 2	Lab 1
DARKO	**0.851**	**0.683**	**0.700**	**0.666**	**0.880**
MMED [6]	0.648	0.563	0.589	0.624	0.683
RNN	0.441	0.322	0.504	0.454	0.651
Logistic	0.517	0.519	0.650	0.657	0.774
Uniform	0.153	0.128	0.154	0.151	0.167

Table 3: **Goal Forecasting Results (Labelled Detections):** Proposed goal posterior achieves best $\overline{P}_{g^*}$ (mean probability of true goal). Methods benefit from better detections.

Method	Home 1	Home 2	Office 1	Office 2	Lab 1
Scene-based	0.438	0.346	0.560	0.238	0.426
Stop-based	0.614	0.395	0.644	0.625	0.709

Table 4: **Visual goal discovery:** Better goal discovery (cf. Table 1) yields better $\overline{P}_{g^*}$. Here, action detection labels are used to isolate performance differences.

recognition. We still use SLAM for localization.

Table 3 summarizes the mean true goal probability for each of the dataset environments. We observe a mean absolute performance improvement of 0.27 by using idealized inputs. Our proposed model continues to perform the best against baselines methods. This performance indicates that as vision-based component technologies improve, we can expect significant improvements in the ability to predict the goals of human activities.

We also measure performance when the action detection is built from ground truth and the goal discovery is built from our described methods. Our expectation is that DARKO with stop-based discovery should outperform DARKO with scene-based based discovery, given the stop-detector's more reliable goal detection performance (Table 1). The results over the dataset are given in Table 4, confirming our expectation.

5.5. Goal Forecasting Performance over Time

In additional to understanding the performance of goal prediction with a single score, we also plot the performance of goal prediction over time. We evaluate the goal forecasting performance as a function of the fraction of time until reaching the goal. In Figure 4, we plot the *mean* probability of the true goal at each fractional time step $\widehat{P}(g^*|\xi_t) = \frac{1}{N}\sum_{n=1}^N P_n(g^*|\xi_{nt})$. Using fractional trajectory length allows for a performance comparison across trajectories of different lengths.

As shown in Figure 4, DARKO exhibits the property of maintaining uncertainty early in the trajectory and converging to the correct prediction as time elapses in most cases. In contrast, the logistic regression, RNN, and MMED perform worse at most time steps. As it approaches the goal, our method always produces a higher confidence in the cor-

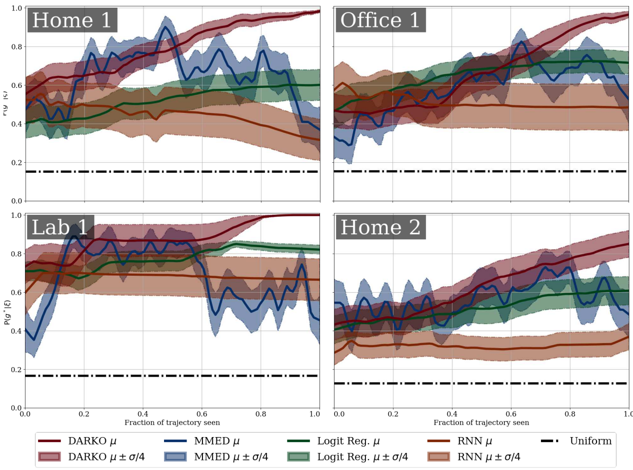

Figure 4: **Goal posterior forecasting over time:** $\widehat{P}_{g^*}$ vs. fraction of trajectory length, across all trajectories. DARKO outperforms other methods and becomes more confident in the correct goal as the trajectories elapse.

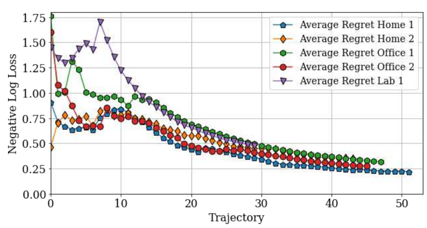

Figure 5: **Empirical regret.** DARKO exhibits sublinear convergence in average regret. Initial noise is overcome after DARKO adjusts to the user's early behaviors.

rect goal with lower variance. We tried `argmax` and Platt scaling [17] to perform multi-class prediction with MMED; `argmax` yielded higher $\overline{P}_{g^*}$, in addition to making $\widehat{P}_{g^*}$ noisier. While the RNN sees many states, its trajectory-centric hidden-state representation may not have enough data to generalize as well as the state-centric baselines.

5.6. Empirical Regret Analysis

We empirically show that our model has no-regret with respect to the best model computed in hindsight under the MaxEntIRL loss function (negative log-loss). In particular, we compute the regret (cumulative loss difference) between our online algorithm and the best hindsight model using the batch MaxEntIOC algorithm [34] at the end of all episodes. We plot the average regret $\frac{\mathcal{R}_t}{t}$ for each environment in the dataset in Figure 5. The average regret of our algorithm approaches zero over time, matching our analysis.

5.7. Evaluation of Trajectory Length Estimates

Our model can also be used to estimate how long it will take a person to reach a predicted goal state. We predict the expected trajectory length as derived in Section 4.1. For the n-th episode, we use the normalized trajectory length prediction error defined as $\epsilon_n = \sum_{t=1}^{T_n} \frac{|\tau_{nt} - \hat{\tau}_{nt}|}{\tau_{nt}}$, where τ_{nt} is the true trajectory length and $\hat{\tau}_{nt}$ (Eq. 6) is the predicted trajectory length. Proper evaluation of trajectory length towards a goal is challenging because our approach must learn valid goals in an online fashion. When a person approaches a new goal, our approach cannot accurately predict the goal because it has yet to learn that it is a valid goal state. As a result, our algorithm makes wrong goal predictions during episodes that terminate in new goal states. If we simply evaluate the mean performance, it will be dominated by the errors of the first episode terminating in a new goal state.

We evaluate median ϵ_n over all N episodes. The median is not dominated by the errors of the first episode toward a new goal. We find most trajectory length forecasts are accurate, evidenced by the median of the normalized prediction error in Table 5. We include a partial-trajectory nearest neighbors baseline (NN). In Lab 1, the median trajectory length estimate is within 6.3% of the true trajectory length.

Statistic	Home 1	Home 2	Office 1	Office 2	Lab 1
Med. % Err.	30.0	34.8	17.3	18.4	6.3
Med. % Err. NN	29.0	33.5	42.9	36.0	35.4
Mean $\lvert\xi\rvert$	20.5	31.0	27.1	13.7	23.5

Table 5: **Trajectory length forecasting results.** Error is relative to the true length of each trajectory. Most trajectory forecasts are fairly accurate.

6. Conclusion

We proposed the first method for continuously modeling and forecasting a first-person camera wearer's future semantic behaviors at far-reaching spatial and temporal horizons. Our method goes beyond predicting the physical trajectory of the user to predict their future semantic goals, and models the user's relationship to objects and their environment. We have proposed several efficient and extensible methods for forecasting other semantic quantities of interest. Exciting avenues for future work include building upon the semantic state representation to model more aspects of the environment (which enables forecasting of more detailed futures), validation against human forecasting performance, and further generalizing the notion of a "goal" and how goals are discovered.

Acknowledgements: We acknowledge the support of NSF NRI Grant 1227495 and JST CREST Grant JPMJCR14E1. The authors thank Drew Bagnell for technical discussion, Katherine Lagree and Gunnar A. Sigurdsson for data collection assistance, and the anonymous reviewers for their helpful comments.

References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Mark Siskind, and S. Wang. Recognize human activities from partially observed videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[4] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *2011 International Conference on Computer Vision*, pages 407–414. IEEE, 2011.

[5] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010.

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

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

[8] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2016.

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

[10] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, volume 2, page 7, 2012.

[11] K. Li and Y. Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1644–1657, 2014.

[12] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

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

[14] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani. A game-theoretic approach to multi-pedestrian activity forecasting. *arXiv preprint arXiv:1604.01431*, 2016.

[15] R. Mur-Artal, J. Montiel, and J. D. Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[16] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012.

[17] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[18] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736. ACM, 2006.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[20] N. Rhinehart and K. M. Kitani. Learning action maps of large environments via first-person vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–588, 2016.

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

[22] M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2737, 2013.

[23] S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

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

[25] H. Soo Park, J.-J. Hwang, Y. Niu, and J. Shi. Egocentric future localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] S. Su, J. P. Hong, J. Shi, and H. S. Park. Social behavior prediction from first person videos. *arXiv preprint arXiv:1611.09464*, 2016.

[27] W. Sun, A. Venkatraman, B. Boots, and J. A. Bagnell. Learning to filter with predictive state inference machines. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1197–1205, 2016.

[28] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[29] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3302–3309. IEEE, 2014.

[30] W. W.-S. Wei. *Time series analysis*. Addison-Wesley publ Reading, 1994.

[31] D. Xie, S. Todorovic, and S.-C. Zhu. Inferring "dark matter" and "dark energy" from videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2224–2231, 2013.

[32] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[33] B. D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.

[34] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 1433–1438, 2008.

[35] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936. IEEE, 2009.