Online Semantic Activity Forecasting with DARKO

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

Abstract

We address the problem of continuously observing and forecasting long-term semantic activities of a first-person camera wearer: what the person will do, where they will go, and what goal they are seeking. In contrast to prior work in trajectory forecasting and short-term activity forecasting, our algorithm, DARKO*, reasons about the future position, future semantic state, and future high-level goals of the camera-wearer at arbitrary spatial and temporal horizons defined only by the wearer's behaviors.* DARKO *learns and forecasts online from first-person observations of the user's daily behaviors. We derive novel mathematical results that enable efficient forecasting of different semantic quantities of interest. We apply our method to a dataset of 5 large-scale environments with 3 different environment types, collected from 3 different users, and experimentally validate* DARKO *on forecasting tasks.*

1. Introduction

Imagine your daily activities. Perhaps you will be at home today, relaxing and completing chores. Maybe you are a scientist, and plan to conduct a long series of experiments in a laboratory. You might work in an office building: you walk about your floor, greeting others, getting coffee, preparing documents, etc. There are many activities you perform regularly in large environments. If a system *understood your intentions* it could help you achieve your goals, or automate aspects of your environment. More generally, an understanding of human intentions would benefit, and is perhaps prerequisite for, AI systems that assist and augment human capabilities.

We present a framework that continuously *forecasts long-term spatial and semantic intentions* (what you will do and where you will go) of a first-person camera wearer. We term our algorithm "Demonstrating Agent Rewards for K-futures Online" (DARKO). We use a first-person camera to meet the challenge of observing the wearer's behavior everywhere. In Figure 1, DARKO forecasts multiple quantities: (1) the user intends to go to the shower (out of all possible destinations in their house), (2) their trajectory through space to this probable goal, and (3) the high expectation that a user will pick up a towel along the way. In Figure 2, DARKO is shown adapting its goal prediction for two sequences that share a similar beginning.

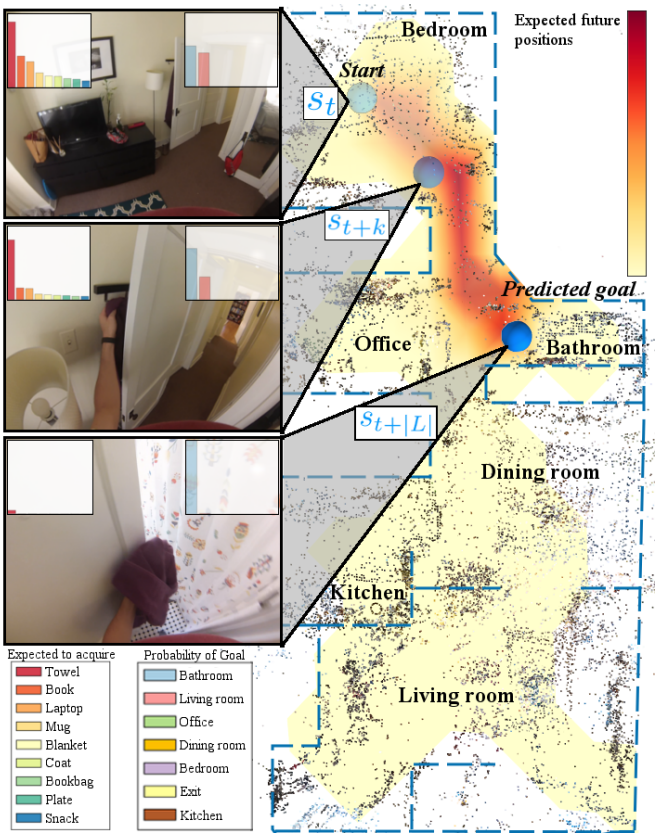


Figure 1: **Forecasting future behavior from first-person video.** The overhead map shows where the person is likely to go, predicted from the first frame. Each s_i refers to the *state* of the user at time i. The histogram insets on each image display the predictions of the person's next long-term semantic goal (inner right) and what objects they will acquire along their way to their goal (inner left). Forecasting is performed and learned continuously as the person behaves regularly.

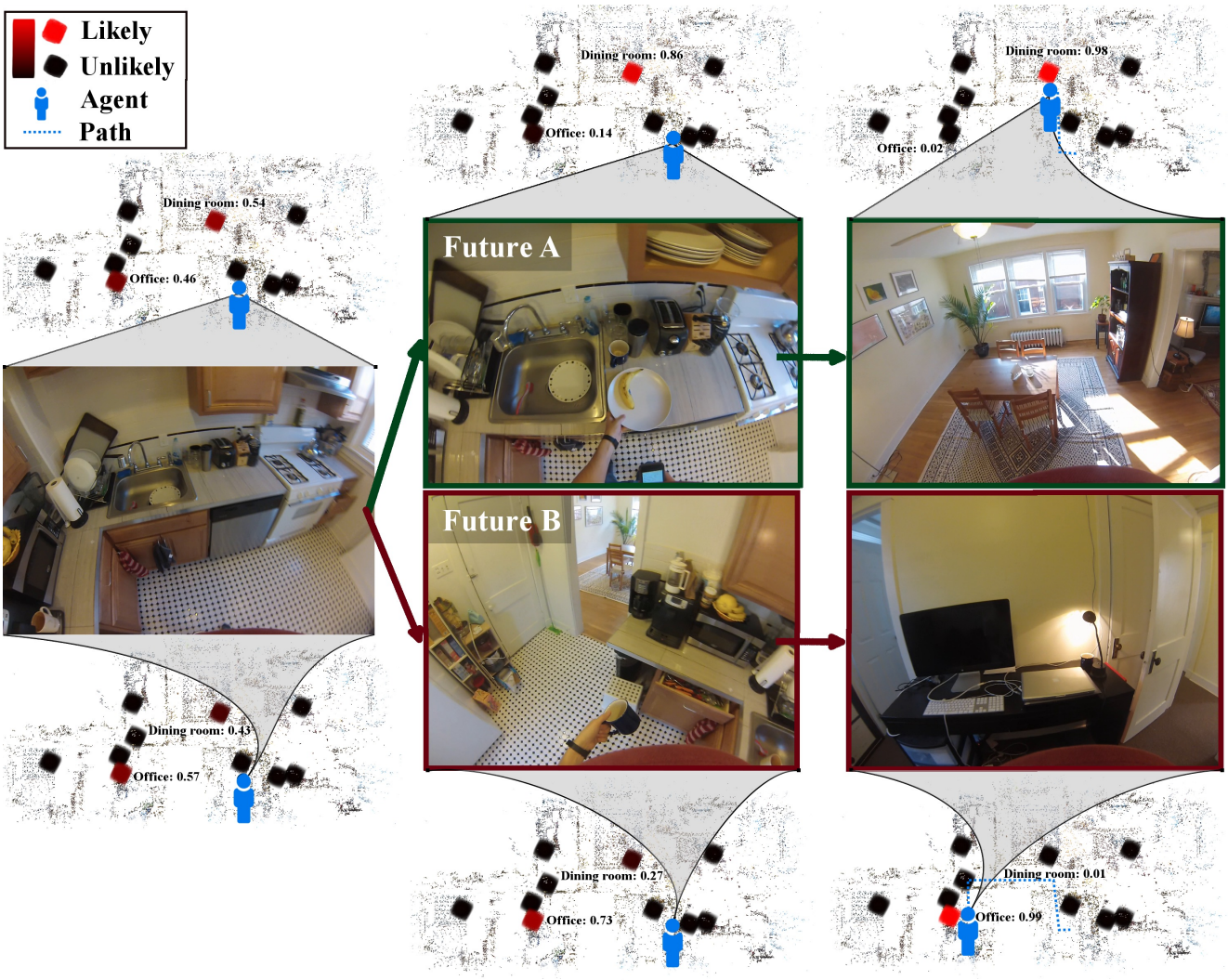


Figure 2: **Goal forecasting over time:** Predictions of future goal for each frame. DARKO's predictions change over time as the person performs activities. In the top sequence, the acquisition of a plate and snack shifts probability mass to the "dining room" goal. In the bottom sequence, the acquisition of a mug shifts probability mass to the "office" goal.

In contrast to the tasks of early event recognition and activity prediction, [6, 8, 16, 19], DARKO predicts activities at an arbitrary time horizon that can adapt as new behaviors are observed: our primary inference task is to predict the future goal. Our model reasons over the spatial and semantic states of the agent and environment, yielding forecasts of the future across space, time, and activities.

Long-term observation across large environments, such as those in Figure 3, necessitates a suitable sensing modality. While it may be reasonable to instrument the environment with networked devices (*e.g.* surveillance cameras), sensing will be constrained to the instrumented space. Additionally, the user has less control over fixed sensors. A first-person camera allows for observation of human actions over large environments that expand as the wearer visits new places. The camera is generally close to the wearer's hands and objects of interaction, thus it is placed at an ideal vantage point from which to understand the wearer's interactions with the world.

Each person has a unique lifestyle and specific environment in which they live and work. Therefore, any feasible approach for learning and predict novel goals should be personalized to an individual's behaviors. A traditional supervised learning approach, *e.g.* a deep network, to build customized activity recognition algorithms can be quite expensive in terms of training data. A per-person online learning approach is more appropriate for the streaming setting, and relaxes the data requirements. We apply the Maximum Entropy Inverse Optimal Control (MaxEntIOC) [24] framework to an online setting, which allows us to continuously

model each agent individually as a near-optimal agent in their own environment. Our online approach allows for *continuous estimation of and adaptation to* the wearer's behaviors, including when they visit new environments.

Our contributions include:

- the first online activity forecasting approach
- the first expansion of the spatial and temporal activity prediction horizons beyond single views, single rooms, and short timescales
- efficient modeling of traversal between many goals
- new mathematical results for forecasting in the MaxEntIOC framework

Our mathematical results include expected subspace forecasting, expected action visitation from a subspace, and expected trajectory length forecasting, described in Section 3.5. Subspace visitation forecasting is especially applicable to our scenario: as we will show, our state space formulation can naturally be partitioned into subspaces that retain important semantic meaning.

2. Related work

We discuss several topics of related prior work. First-person vision encompasses a range of problems, including activity and object detection. Early event and early action recognition are the tasks of predicting the current ongoing activity as soon as possible, whereas the problem of action forecasting involves prediction a future activity for which there is yet no direct evidence. Trajectory forecasting produces predictions of where a person will go in the future. Maximum entropy IOC has been employed to learn goal-directed models for humans.

First-person vision: The first-person vision tasks most relevant to our work are those that undertake the challenging problems of object and activity recognition from the first-person view point [4, 10, 12, 15, 17]. In [5], the authors note that actions are identifiable by how they change the state of the environment, similar to how we seek to model the state of the user in their environment. These recognition tasks *identify the present nature* of the environment or user, complementary to our task of *inferring the future nature*.

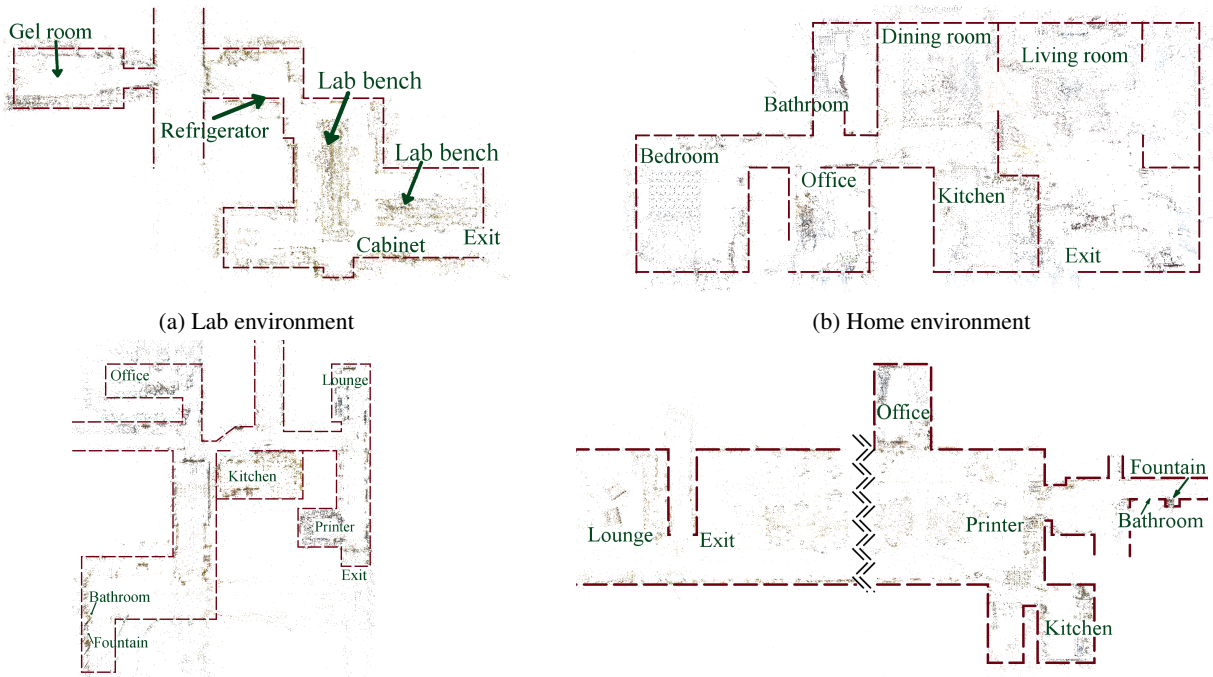
Early recognition and action forecasting: In [6, 16], the tasks are to recognize an unfinished event or activity. In [9], a hierarchical structured SVM is employed to forecast actions about a second in the future, and [19] demonstrates a semi-supervised approach for forecasting actions a second into the future. Other related short-term prediction methods include [3, 11, 20]. These methods are constrained to deal with activities that occur under small or fixed time windows. However, many human activities are performed over different time scales and longer time scales. Our framework allows for prediction of different activities at different time scales. In the extreme case, a person may not reach their goal for hours (*e.g.*, walking long distances), yet our framework would allow for prediction of their future activities in this scenario. Additionally, these methods do not adapt to the *specific environment of the agent*: they are not aware of the larger space in which the activities are situated and history of the user's behavior in that space.

Trajectory forecasting and Maximum Entropy IOC: [18] 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. Multiple human trajectory forecasting from a surveillance camera was investigated by [2, 13]. Another trajectory forecasting approach uses demonstrations observed from a bird's-eye view of a scene to infer latent goal locations [21]. The authors of [24] introduced the maximum entropy approach for inverse reinforcement learning, and modelled behavior of taxi-cab drivers in a road network to perform taxi trajectory forecasting. [25] demonstrated pedestrian trajectory forecasting in a single room using the MaxEntIOC framework, tracking the person's movements with laser range finders. The work of [7] brought the MaxEntIOC approach into the realm of computer vision and forecasted the future trajectory of a person from a 3rd-person surveillance image, using discrete locations on the ground plane as the state space.

Our approach introduces several novelties over existing forecasting approaches. We go beyond physical trajectory forecasting and produce forecasts of semantic goals. DARKO continuously expands the state space and learns a model as the person demonstrates behaviors, enabling long-term modeling for arbitrary users. Our method reasons about futures far outside of the view of a single frame and adapts to each individual's environment. We model more aspects of goal-directed behavior than prior MaxEntIOC work, including the sequential nature of reaching goals, and the interaction with objects. Finally, we contribute several new inference methods to the MaxEntIOC framework.

3. Approach

We learn a model of the camera-wearer's (hereafter "agent" or "user") behavior that facilitates forecasting of future activities. We assume the agent will behave near-optimally: they achieve their goals in a reasonably efficient manner. Given demonstrations from the agent, the task of *inverse optimal control* (IOC) is to recover a *reward function* (or cost function) that explains the agent's behavior as near-optimal in a Markov Decision Process (MDP). However, there is generally more than one choice of reward function that explains an agent's behavior, among other ambiguities [1]. In [24], the principal of Maximum Entropy is employed to resolve these ambiguities, which results in a model that exhibits minimal preference between behaviors



(a) Lab environment

(b) Home environment

(c) Office environment 1

(d) Office environment 2

Figure 3: **Environments:** SLAM points and stylized blueprints of several environments from our dataset are shown. Inter-environment distances not to scale. Doubly-dashed lines in Figure 3d indicate a shortened environment for layout clarity.

that obtain equivalent rewards. Thus, uncertainty is preserved at the beginning of a demonstrated behavior: possible future goals receive approximately equal probabilities. We revisit this property later.

An MDP-based approach requires an estimate of the *state* of a user (described further in Section 3.1). Our state includes a spatial component, thus, we must continuously estimate where the person is. We make use of a vision-based SLAM algorithm, [14], to localize the agent in space. To identify the end of a single demonstration (a *trajectory*), we must detect when a person has arrived at a goal state. Our state also includes a semantic component of object possession: what objects a person is carrying.

The layout of Section 3 is as follows. We introduce our terminology and framework in Section 3.1. We present our main algorithm, DARKO, in Section 3.2. In Section 3.3, we show how we construct our state space to enable modeling an agent seeking goals sequentially. We overview known results of goal posterior prediction in Section 3.4, and derive new results for subspace visitation and trajectory forecasting in Section 3.5.

3.1. Terminology

Our Markov Decision Process framework is:

$$\text{MDP} = (\mathcal{S}, \mathcal{A}, T(\cdot,\cdot), R_\theta(\cdot,\cdot)), \tag{1}$$

where $\mathcal{S}$ is the expanding state space, which is the set of states an agent can visit, and $\mathcal{A}$ is expanding action space comprised of independent movement and activity transitions. $T(s,a) \mapsto s'$ is a deterministic transition function. $R_\theta(s,a) = \theta^T f(s,a)$ is an instantaneous linear reward function of taking an action at a state, learned online from demonstrations with features $f(s,a)$ and parametrized by θ. A policy $\pi(a|s)$ gives the probability of taking action a while in state s. A trajectory ξ is composed of state-action tuples: $\xi = [(s_0,a_0),\ldots,(s_N,a_N)]$. We use $\xi_{0\to t}$ to indicate a trajectory starting at time 0 and ending at time t.

3.2. Algorithm

DARKO, shown in Algorithm 1, continuously forecasts future activities. Frames are obtained from a first-person camera (the NEWFRAME function), and a SLAM algorithm is used to track the user's location. Part of the wearer's semantic state is obtained from an action detection algorithm, ACTIONDET. Feedback is obtained after every detected arrival at a goal, providing a new demonstration trajectory, ξ. This feedback is used to update the current policy, π. Thus, the policy is continually fit to incorporate new information about a person's behaviors. The policy is a model of the wearer that can be forward simulated to obtain predictions of the person's future behavior.

The FORECAST function, parameterized by the current policy π, runs trajectory inference (Algorithm 11.4 from

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

func DARKO(SLAM, ACTIONDET, GOALDET, T)
  $s \leftarrow \mathbf{0}, \mathcal{S} = \{\}, \xi = [], \mathcal{G} = \{\}, \theta = \mathbf{0}$
  while True **do**
    frame $\leftarrow$ NEWFRAME()
    pose $\leftarrow$ SLAM.TRACK(frame)
    $a \leftarrow$ ACTIONDET(pose, frame)
    $\xi \leftarrow \xi \oplus (s, a), \mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$
    $s \leftarrow T(s, a)$
    FORECAST$_\pi(\xi)$
    is_goal $\leftarrow$ GOALDET(s, pose, frame, $\mathcal{G}$)
    if is_goal **then**
      $\mathcal{G} \leftarrow \mathcal{G} \cup \{s\}$
      $\pi \leftarrow$ FIT($\theta, \mathcal{S}, T, \xi, \mathcal{G}$), $\xi = []$
      $s \leftarrow T(s, a =$ at_goal$)$
    end if
  end while
end func

Algorithm 2 Policy fitting

func FIT($\theta, \mathcal{S}, T, \xi, \mathcal{G}; \lambda$)
  $\bar{f}_i = \sum_{(s,a)\in\xi} f(s, a)$
  $\pi \leftarrow$ SOFTVALUEITERATION($\theta, \mathcal{S}, \mathcal{G}, T$)
  $\hat{f}_i \leftarrow$ EXPECTEDFEATURE($\mathcal{S}, \pi, T$)
  $\theta \leftarrow \theta - \lambda(\bar{f}_i - \hat{f}_i)$
  return π
end func

[23]) to compute Equations 2 and 3, and additionally computes Equations 4, 6, and 8 for various queries. The primary quantity of interest is the *goal posterior* of Equation 2: $P(g|\xi_{0\to t})\forall g \in \mathcal{G}$. This represents probabilistically *what goal the user seeks given their current trajectory*. Note that goal states can share locations if their semantic states differ. The FIT function, shown in Algorithm 2, performs gradient descent on the current parameters of the policy, employing the soft value iteration and expected feature algorithms of [23] to calculate the gradient of the reward function and update its parameters.

3.3. State space construction

Consider the following scenario: an agent starts in state s_a and travels to goal state s_g. For example: a person travels from an office to the kitchen to get coffee, and then travels back from the kitchen to the office: $s_a \to s_g, s_g \to s_a$. The value of s_g should be higher than the value at all other states to encourage the agent to travel from s_a to s_g, yet, after arriving at s_g, the agent moves on to another goal. Under the simple environment and value function V unaware of the previous goal, moving from s_g to s_a would be highly suboptimal (*i.e.* $V(s_g) \gg V(s_a)$): the optimal action is to remain at s_g. Therefore, we augment the state space with the *index of the previous goal*: after arriving at s_g, the environment (T) transitions the agent to a new state, s'_g, identical to s_g except for its previous goal index. Value iteration propagates from goal states and does not spread across this environmental barrier, decoupling the values of s_g and s'_g.

We model the objects the person currently has in possession. The objects a person picks up and uses for various activities are strong cues for their behaviors [4], and as discussed later, this formulation enables inference of the objects a person will acquire and release in the future. With N objects and K possible goals, a state in our state space is written

$$s = [x, y, z, g_1, \dots g_K, o_1 \dots, o_N],$$

where x, y, z are the discretized 3D components of position, $g_i = 1$ if the person last arrived at goal category i, and $o_j = 1$ if the person has object j in their possession.

3.4. Future goal prediction

In MaxEntIOC with deterministic transitions, the posterior over goals given a partial trajectory is defined in Equation 2:

$$P(G|\xi_{0\to t}) \propto P(G)e^{V_{s_t}(G)-V_{s_1}(G)}, \tag{2}$$

where $V_{s_i}(G)$ is inverted value function of G obtained by performing soft-value iteration with s_i as the goal state (s_0 and s_t are the starting and current states). Roughly, the probability of a goal is exponentially proportional to the progress made towards that goal under the value function. Online goal forecasting thus requires running soft-value iteration twice.

3.5. Future state and subspace prediction

Goal forecasting predicts the long-term activity of a user. We are also interested in *what states* a person will reach and *what actions* they will perform along their way to the goal. The posterior expected count of future visitation to a state s_x (Equation 11.4 in [23]) is defined as

$$D_{s_x|\xi_{0\to t}} \triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau = s_x)\right]. \tag{3}$$

This quantity represents how frequently the agent is expected to visit each state in the future given the current trajectory, $\xi_{0\to t}$, under the maximum causal entropy distribution, $P(\xi_{t+1\to T}|\xi_{0\to t})$, where I is the indicator function. The maximum causal entropy distribution yields the probability of a future trajectory given the current trajectory.

The posterior expected count of future states in Equation 3 can be efficiently computed using Algorithm 11.4 from [23]. We are interested in generalizing this result from

a single state to a *subset of states*. By exploiting the structure of our state space, we can construct subsets of the state space that have an interesting and relevant semantic meaning, such as "having an object o_i" or "all states closest to goal k with $\mathcal{O}$ set of objects."

Our interest now is in the quantity in Equation 4, the expected count of visitation to a subset of states $\mathcal{S}_p$ satisfying some property p.

$$D_{\mathcal{S}_p|\xi_{0\to t}} \triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau \in \mathcal{S}_p)\right] \tag{4}$$

We show how to rewrite this in terms of Equation 3:

$$\begin{aligned} D_{\mathcal{S}_p|\xi_{0\to t}} &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{s_x\in\mathcal{S}_p}\sum_{\tau=t+1}^{T} I(s_\tau = s_x)\right] \\ &= \sum_{s_x\in\mathcal{S}_p} \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau = s_x)\right] \\ &= \sum_{s_x\in\mathcal{S}_p} D_{s_x|\xi_{0\to t}}. \end{aligned} \tag{5}$$

Thus, we can efficiently compute the posterior expected visitation count to a subset of states.

To answer questions like "what is the likelihood of acquiring a specific object along the way to the goal" (*e.g.* acquiring a coat before going to the exit), we must consider the subspace of states with the action of interest available. Also relevant is the policy's probability of performing an action at each state in the subspace. We formulate the posterior expected visitation count of performing an action a_y immediately after arriving in a subspace, $\mathcal{S}_p$, as

$$D_{a_y,\mathcal{S}_p|\xi_{0\to t}} = \sum_{s_x\in\mathcal{S}_p} \pi(a_y|s_x) D_{s_x|\xi_{0\to t}}. \tag{6}$$

Equation 6 is the straightforward extension of $D_{a_{s_x},s_y}$ from [23] to the subspace case. We can push this result further to compute expected transition counts of a joint over subsets of both actions ($\mathcal{A}_y$) and states ($\mathcal{S}_p$) in Equation 7,

$$D_{\mathcal{A}_y,\mathcal{S}_p|\xi_{0\to t}} = \sum_{a_y\in\mathcal{A}_y}\sum_{s_x\in\mathcal{S}_p} \pi(a_y|s_x) D_{s_x|\xi_{0\to t}}, \tag{7}$$

by following a similar derivation to that of Equation 5. Thus, under our state-space model, we can efficiently perform inference to answer questions like "what is the expected number of times the person will acquire *any* object (subset of actions) in the next k steps, given their current state?"

Given the user's state, how long will it take them to reach their next goal? The answer tells us more about the user's intentions than their goal and activities alone. We consider the *expected trajectory length* in Equation 8. The notation $|\xi|$ is used to indicate the count of states in trajectory ξ.

$$L_{\xi_{t+1\to T}|\xi_{0\to t}} \triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})} |\xi_{t+1\to T}| \tag{8}$$

This is nontrivial to compute at first glance, as it involves the causal maximum entropy distribution over future trajectories. Consider evaluating $D_{\mathcal{S}_p|\xi_{0\to t}}$ from Equation 5 by setting $\mathcal{S}_p = \mathcal{S}$, that is, by considering the expected future visitation count to the entire state space. Then,

$$\begin{aligned} D_{\mathcal{S}|\xi_{0\to t}} &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau \in \mathcal{S})\right] \\ &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} 1\right] \\ &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})} |\xi_{t+1\to T}| \\ \sum_{s_x\in\mathcal{S}} D_{s_x|\xi_{0\to t}} &= L_{\xi_{t+1\to T}|\xi_{0\to t}} \end{aligned} \tag{9}$$

Therefore, we can compute the expected future trajectory length by summing the expected state frequencies over the entire state space. We use this identity to continuously estimate the agent's future trajectory length.

4. Experiments

We introduce our dataset in Section 4.1, and evaluate our primary inference task of online goal forecasting in Section 4.2, comparing against several baselines related to DARKO. In most experiments, we employ detectors derived from the activity and goal *labels*. Although we also analyze performance with a goal-detection approach in Section 4.3, our focus is not to engineer the detectors that serve as input to DARKO. Additionally, we evaluate online trajectory length forecasting in Section 4.4 using the results derived in Section 3.5.

4.1. Dataset

We collected a dataset of sequential goal-seeking behavior in several different environments under the following high-level scenarios "home," "office," and "laboratory." The users recorded a series of activities that naturally occur in each scenario. Our dataset is comprised of 5 environments, from 3 different users, and includes over 250 activities with 23 objects, 17 different goal types, at least 6 goals per environment, and about 200 different high-level activities. In each environment, the user recorded 3-4 long sequences of high-level activities, where each sequence represents a full day of behavior, temporally compressed to be sequential (very few large delays between activities). Thus, our dataset represents over 15 days of recording. Large delays between activities (e.g. sitting on a couch for a prolonged period of time) are still handled by our framework, as the user's state will simply be static during this delay.

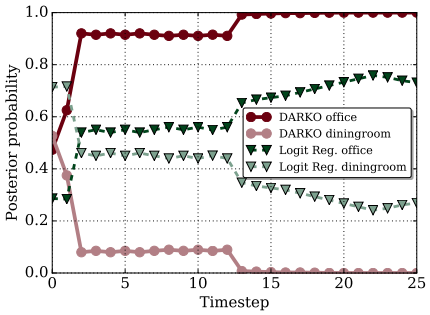
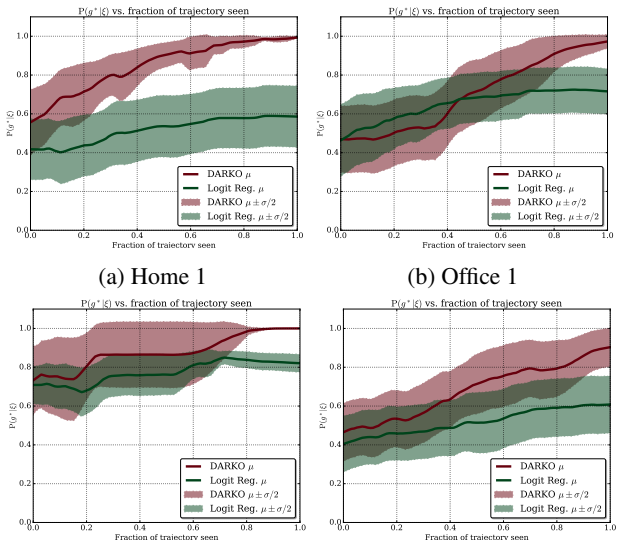


Figure 4: **DARKO vs. Logistic Regression goal posteriors for an example trajectory**. Initially, uncertainty is maintained by DARKO's MaxEntIOC forecasting, as the user could go to either goal. Logistic regression is overconfident in its initial (incorrect) prediction and is underconfident in its final prediction, as it does not explicitly have information about the user's history. As the user approaches the true goal, DARKO's goal posterior is very confident in its prediction - the user is very unlikely to go to the office given the majority of the trajectory.



(a) Home 1 (b) Office 1

(c) Lab 1 (d) Home 2

Figure 5: **Goal posterior forecasting over time:** mean $P(g^*|\xi)$ vs. fraction of trajectory length

	Home 1	Home 2	Office 1	Office 2	Lab 1
DARKO	**0.851**	**0.683**	**0.700**	**0.666**	0.880
DARKO$_{R(s)}$	0.735	0.574	0.581	0.549	**0.892**
Logit Reg.	0.517	0.519	0.650	0.657	0.774
Tabular	0.578	0.403	0.395	0.619	0.826
Uniform	0.153	0.128	0.154	0.151	0.167

Table 1: **Goal posterior results.** AUCs of $P(g^*|\xi)$ vs. percentage of elapsed trajectory produced by DARKO and the baselines.

4.2. Goal forecasting

Goal forecasting baselines: We introduce several online methods for producing a posterior over future goals. Our first baseline is a *uniform posterior over possible goals*. This gives equal weight to all known goals for the current frame. The next baseline is the *normalized tabular counts* baseline. For K goal types, a $K \times K$ table, U, tracks the number of visits to a goal state when starting from a start state. A visit to goal j after starting at goal i is recorded as incrementing u_{ij}, and predictions are formed by obtaining the current starting state, and producing a posterior by normalizing across u_i^T. This baseline makes the reasonable assumption that starting states are correlated with goal states. A final baseline is *logistic regression from state features to goals*, retrained online with the history of demonstration trajectories as they become available, which is a Follow-the-Leader online learning approach. This baseline has access to the same information as DARKO, but only reasons about the current state (and does not use the full history of the trajectory). Each baseline requires the state tracking and goal detection components of DARKO. The baselines are limited in that they are not immediately applicable to other inference tasks, such as computing the posterior probabilities of trajectories, computing the estimated future trajectory length, forecasting visitation to future intermediate states and actions, etc.

Results and maintaining early uncertainty: A user may be in a state and go to one of several goals (*e.g.* Figure 2), but generally takes a near-optimal path to the goal they have in mind. Our framework allows us to both maintain early uncertainty and harness user history as they approach the goal. Early uncertainty results from encouraging the policy to match observed behaviors that contain forking paths. We evaluate the quantity $P(g^*|\xi)$, which is a method's output probability of the *true* goal, g^* given a partial trajectory. We show in Figure 4 an example of DARKO maintaining early uncertainty and converging to the correct answer as the user approaches the goal. In contrast, logistic regression has no such guarantees and fails in in this example. We plot statistics of $P(g^*|\xi)$ vs. fraction of trajectory seen for every trajectory in each environment in Figure 5, and further summarize these results in Table 1 by computing the Area Under the Curve for each: AUC$_{P(g^*|\xi)}$. We include in Table 1 a variant of our approach, DARKO$_{R(s)}$, which does not use features of actions in the reward function (only features of state). This helps us measure the importance of explicitly modelling future actions, and we find that including both components in the feature is almost always better. We find that in all cases, DARKO outperforms the best baseline, and in almost all cases, DARKO exhibits the property of maintaining uncertainty early in the trajec-

Figure 6: **Goal forecasting examples:** A temporal sequence of goal forecasting results is shown in each row from left to right, with the forecasting destination icons, labels, and probabilities inset sorted by probability. *Top*: the scientist acquires a sample to boil in the gel room. *Middle*: the user acquires a textbook and travels to the lounge to read. *Bottom*: the user leaves their apartment.

tory and converging to the correct prediction as time elapses ($\lim_{\frac{t}{T}\to 1} P(g^*|\xi_{0\to t} \approx 1)$). Visual examples of these results are shown in Figure 6.

4.3. Forecasting with goal detection

We analyze the results of DARKO with a scene-classifier-based goal detection method. We reiterate that our focus is not to engineer the required goal and low-level activity detectors. We employ the off-the-shelf scene classifier from [22] to build a temporal window detector on the outputs of the scene classifier. We construct a mapping from scene classification classes to goal types, and compute the windowed average of the probability of each goal type. Using K per-goal averages, we employ K thresholds for goal detection. With a larger dataset of environments, the per-goal thresholds could be tuned automatically. DARKO with a scene-classifier-based goal detector performed best on the Home 1 environment, with an overall detector accuracy of 93.1% and only a 3.1% loss in performance. These results are presented in Table 2. We found in one case (Office 1) a slight increase in overall AUC due to the detector's high accuracy for the most visited goal type in that environment ("office"), and lower accuracy on other goals. More sophisticated methods could be employed for goal detection, for instance, a combination of object and scene classification could be employed to handle goals that are better described by specific objects rather than by specific rooms.

	Home 1	Home 2	Office 1	Office 2	Lab 1
% orig. AUC	96.9	89.2	101.0	89.9	75.0
Detector acc.	93.1	23.6	61.5	48.5	32.0

Table 2: **Goal posterior results with goal detection.** % of original $AUC_{P(g^*|\xi)}$ after integrating a scene-classifier-based goal detector, and goal detector overall accuracy.

	Home 1	Home 2	Office 1	Office 2	Lab 1
Median % Err.	9.0	12.1	3.0	3.4	0.4
Mean % Err.	38.6	105.2	85.6	22.0	5.8
$\frac{1}{N}\sum_{i=1}^{N}\|\xi\|$	20.5	31.0	27.1	13.7	23.5

Table 3: **Trajectory length forecasting results.** Error is relative to the true length of each trajectory. Most trajectory forecasts are fairly accurate (row 1), but the mean % error (row 2) is high, primarily due to outlier trajectories in the failure case of visiting unexplored goals.

4.4. Expected trajectory length forecasting

We evaluate the trajectory length predictions by applying the results derived in Section 3.5. We find that most trajectory forecasts are accurate, and show the median and mean % errors in Table 3. The main source of error is in forecasting trajectories to new goals. When a user demonstrates a *new behavior*, the trajectory length forecast can be arbitrarily bad, as the system is constrained to predict to all known states. For example, if there are only two known goals, the user could walk from the first goal in the opposite direction of the second goal, causing the trajectory forecast and its error to grow as a function of time, as DARKO would predict the user will go to the second goal.

5. Conclusion

We have proposed a 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 the objects in 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), and further generalizing the notion of a "goal" and how goals are discovered.

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. 3

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. 3

[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. 3

[4] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *2011 International Conference on Computer Vision*, pages 407–414. IEEE, 2011. 3, 5

[5] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 3

[6] M. Hoai and F. De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014. 2, 3

[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. 3

[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. 2

[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. 3

[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. 3

[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. 3

[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. 3

[13] 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. 3

[14] 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. 4

[15] 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. 3

[16] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011. 2, 3

[17] 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. 3

[18] 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. 3

[19] 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. 2, 3

[20] 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. 3

[21] 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. 3

[22] 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. 8

[23] B. D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010. 5, 6, 10

[24] 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. 2, 3

[25] 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. 3

A. Proof of Equations 6 and 7

Equation 6 gives the posterior expected visitation count of performing an action a_y immediately after arriving in a subspace S_p. We use the definition and expansion of the expected action visitation count from [23] in Equations 10, 11.

$$D_{a_y,s_x|\xi_{0\to t}} \triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau = s_x)I(a_\tau = a_y)\right] \tag{10}$$

$$D_{a_y,s_x|\xi_{0\to t}} = \pi(a_y|s_x)D_{s_x|\xi_{0\to t}} \tag{11}$$

Our definition of the posterior expected action subspace visitation count is given in Equation 12:

$$\begin{aligned} D_{a_y,\mathcal{S}_p|\xi_{0\to t}} &\triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau \in \mathcal{S}_p)I(a_\tau = a_y)\right] \qquad (12)\\ &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{s_x\in\mathcal{S}_p}\sum_{\tau=t+1}^{T} I(s_\tau = s_x)I(a_\tau = a_y)\right] \\ &= \sum_{s_x\in\mathcal{S}_p} \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau = s_x)I(a_\tau = a_y)\right] \\ &= \sum_{s_x\in\mathcal{S}_p} D_{a_y,s_x|\xi_{0\to t}} \\ &= \sum_{s_x\in\mathcal{S}_p} \pi(a_y|s_x)D_{s_x|\xi_{0\to t}} \qquad (13) \end{aligned}$$

Equation 6 gave the expected transition count from a subspace of states to a subspace of actions, which is defined in Equation 14.

$$\begin{aligned} D_{a_y,\mathcal{S}_p|\xi_{0\to t}} &\triangleq \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau \in \mathcal{S}_p)I(a_\tau \in \mathcal{A}_y)\right] \qquad (14)\\ &= \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{a_y\in\mathcal{A}_y}\sum_{s_x\in\mathcal{S}_p}\sum_{\tau=t+1}^{T} I(s_\tau = s_x)I(a_\tau = a_y)\right] \\ &= \sum_{a_y\in\mathcal{A}_y}\sum_{s_x\in\mathcal{S}_p} \mathbb{E}_{P(\xi_{t+1\to T}|\xi_{0\to t})}\left[\sum_{\tau=t+1}^{T} I(s_\tau = s_x)I(a_\tau = a_y)\right] \\ &= \sum_{a_y\in\mathcal{A}_y}\sum_{s_x\in\mathcal{S}_p} D_{a_y,s_x|\xi_{0\to t}} \\ &= \sum_{a_y\in\mathcal{A}_y}\sum_{s_x\in\mathcal{S}_p} \pi(a_y|s_x)D_{s_x|\xi_{0\to t}} \qquad (15) \end{aligned}$$

B. Goal-detection performance plots

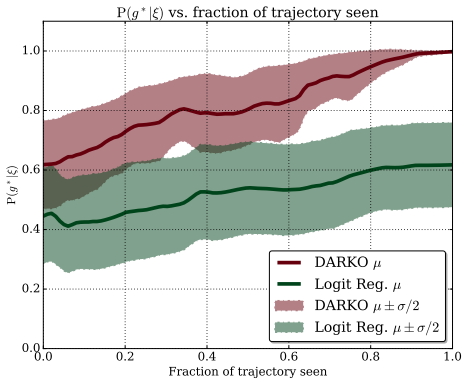
Figures 7 and 8 show the results of using a scene-classification-based goal detector.

B.1. Office 2 performance (no goal detector)

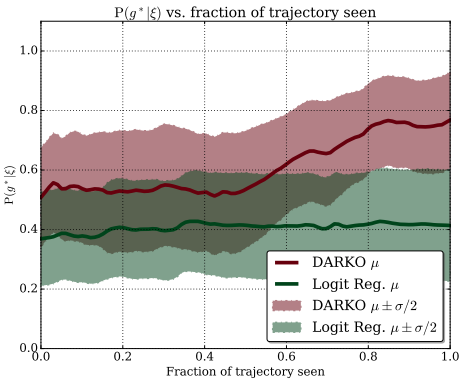
Figure 9 was not shown in the main body for space reasons.

C. Negative-Log Likelihoods

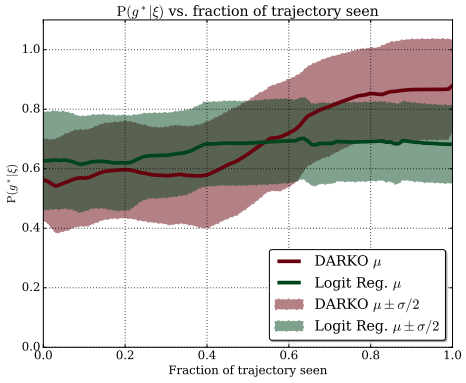
The negative log-likelihood (NLL) measures the extent to which the demonstrated behavior (each ξ) can be explained by the estimated policy π. Low NLL scores indicate the policy matches the demonstrated behavior well. We compute the



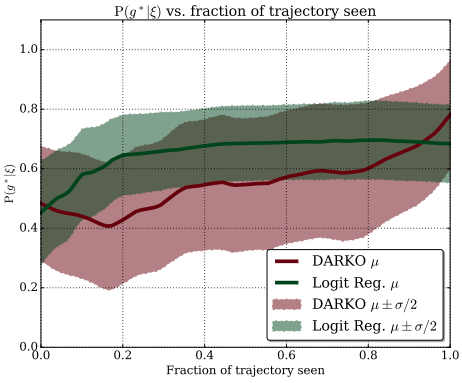
(a) Home 1 with goal detector



(b) Home 2 with goal detector

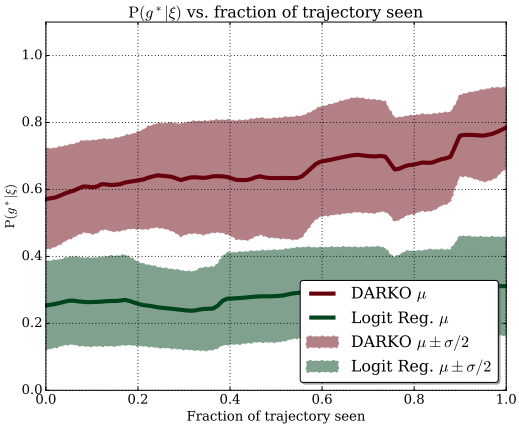


(c) Office 1 with goal detector



(d) Office 2 with goal detector

Figure 7: mean $P(g^*|\xi)$ vs. fraction of trajectory length with CNN-based goal detectors



(a) Lab 1 with goal detector

Figure 8: mean $P(g^*|\xi)$ vs. fraction of trajectory length with CNN-based goal detector

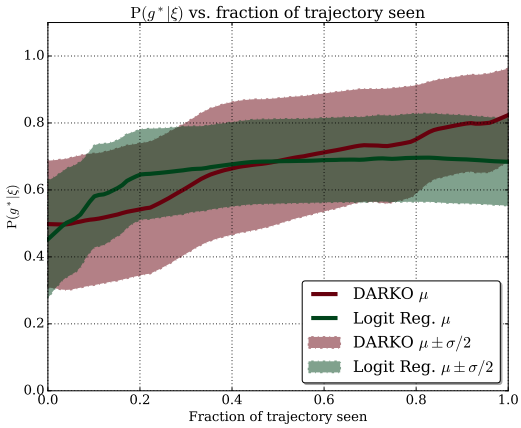


Figure 9: Office 2 with no goal detector

trajectory length-normalized average negative log-likelihood for all demonstrations in each environment as another measure of model performance, and show the results in Table 4. This quantity is computed as:

$$NLL = -\frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{|\xi_i|}\sum_{j=0}^{|\xi_i|-1}\log(\pi(\xi_{ija}|\xi_{ijs})), \tag{16}$$

where ξ_{ija} is the action taken for trajectory i at timestep j, ξ_{ijs} is the state of trajectory i at timestep j, and π is the final policy.

	Home 1	Home 2	Office 1	Office 2	Lab 1
Avg NLL.	1.235	1.836	1.050	1.129	0.584

Table 4: Average NLL for demonstrations in each environment.

D. Future state visitation examples

See Figure 10 for example visualizations of the expected future visitation counts. In order to visualize in 3 dimensions, we take the max visitation count across all states at each position. In rows 1, 3, and 4, a single demonstration is shown, which adapts to the agent's trajectory (history). In row 2, the future state distribution drastically changes after each time the agent reaches a new goal.

E. Value function examples

See Figure 11 for example visualizations of the value function over time. Note 1) the state space size changes, and 2) that the value function changes over time, as the component of state that indicates the previous goal affects the value function.

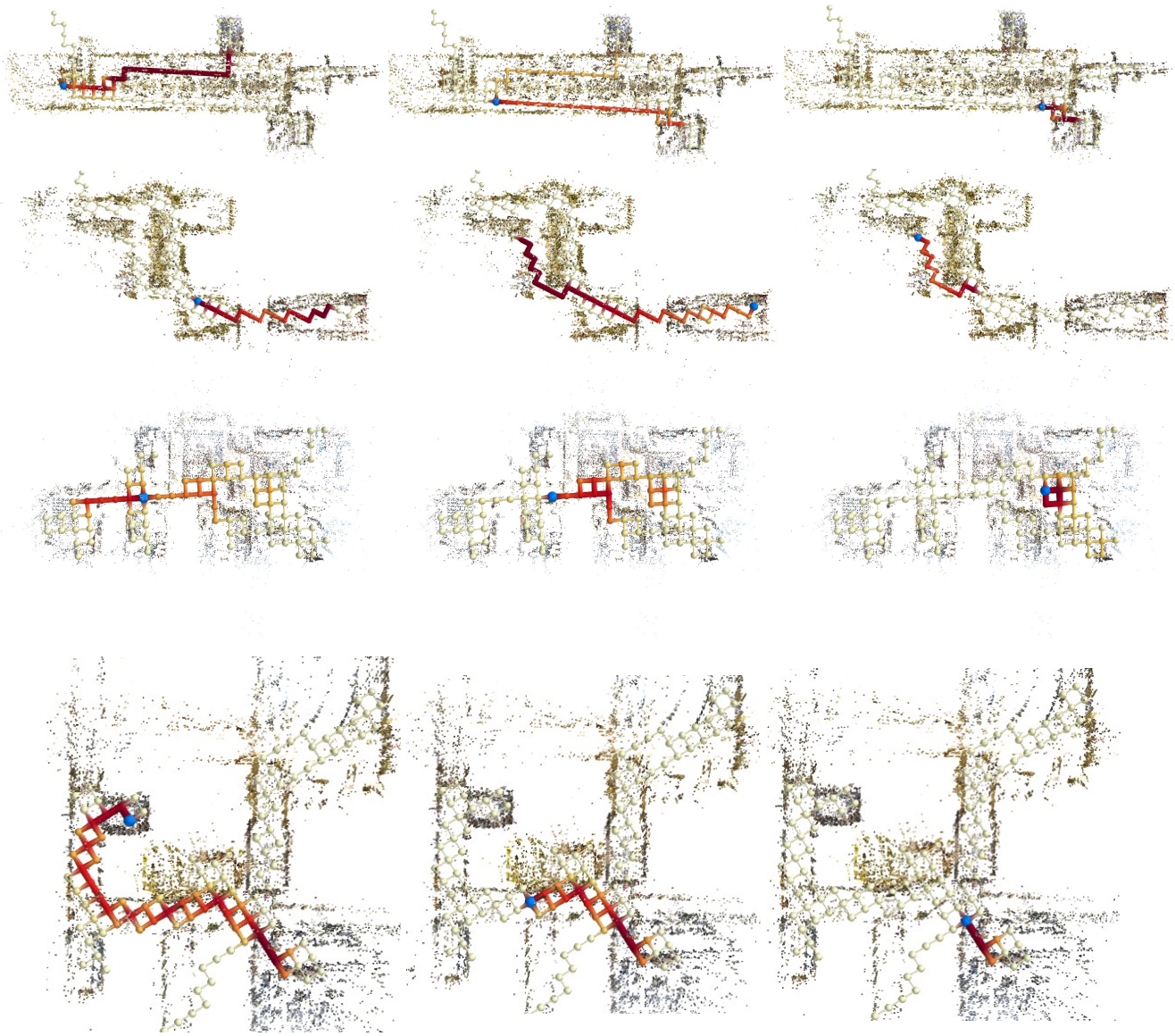


Figure 10: Future state visitation predictions changing as the agent (blue sphere) follows their trajectory. The state visitations are projected to 3D by taking the max over all states at each location. The visualizations are, by row: Office 1, Lab 1, Home 1., Office 2

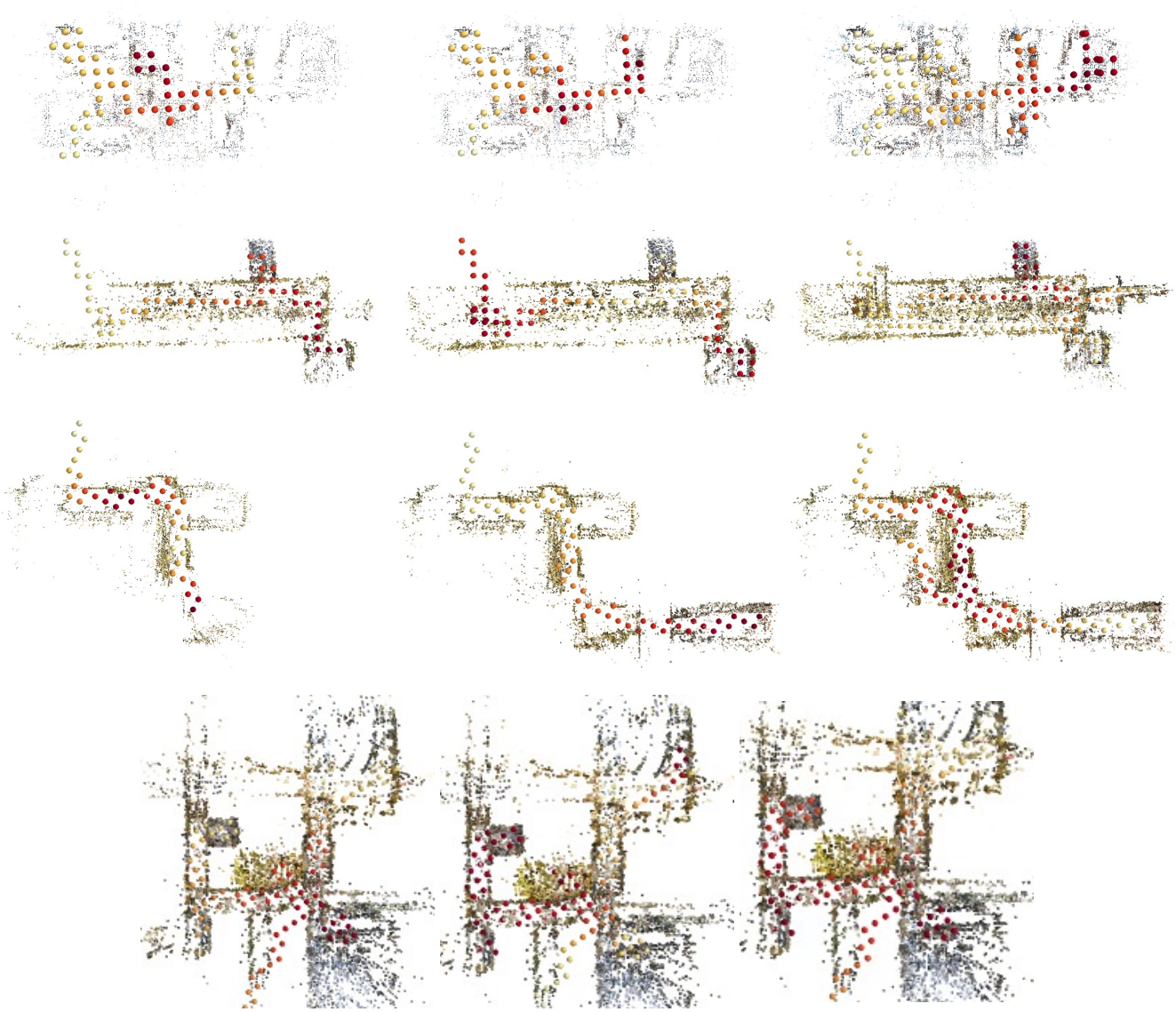


Figure 11: Projections of the value function ($V(s)$) for environments as time elapses (left to right). The state space expands as the user visits more locations. For each position, the maximum value (across all states at that position) is displayed: $\max_{s \in \mathcal{S}_x} V(s)$. From top to bottom, the environments are Home 1, Office 1, Lab 1.