

# Intent Prediction and Trajectory Forecasting via Predictive Inverse Linear-Quadratic Regulation

**Mathew Monfort**

Department of Computer Science  
University of Illinois at Chicago  
Chicago, IL 60607  
mmonfo2@uic.edu

**Anqi Liu**

Department of Computer Science  
University of Illinois at Chicago  
Chicago, IL 60607  
aliu33@uic.edu

**Brian D. Ziebart**

Department of Computer Science  
University of Illinois at Chicago  
Chicago, IL 60607  
bziebart@uic.edu

## Abstract

To facilitate interaction with people, robots must not only recognize current actions, but also infer a person’s intentions and future behavior. Recent advances in depth camera technology have significantly improved human motion tracking. However, the inherent high dimensionality of interacting with the physical world makes efficiently forecasting human intention and future behavior a challenging task. Predictive methods that estimate uncertainty are therefore critical for supporting appropriate robotic responses to the many ambiguities posed within the human-robot interaction setting.

We address these two challenges, high dimensionality and uncertainty, by employing predictive inverse optimal control methods to estimate a probabilistic model of human motion trajectories. Our inverse optimal control formulation estimates quadratic cost functions that best rationalize observed trajectories framed as solutions to linear-quadratic regularization problems. The formulation calibrates its uncertainty from observed motion trajectories, and is efficient in high-dimensional state spaces with linear dynamics. We demonstrate its effectiveness on a task of anticipating the future trajectories, target locations and activity intentions of hand motions.

## Introduction

There has been an increasing desire for co-robotic applications that situate robots as partners with humans in cooperative and tightly interactive tasks (Trafton et al. 2013; Strabala et al. 2013; Kidokoro et al. 2013). Unlike previous generations of human-robot interaction applications, which might need to respond to recognized human behavior (Pineau et al. 2003), co-robotics requires robots to act in anticipation of future human behavior to realize the desired levels of seamless interaction.

Depth cameras, like the Microsoft Kinect, have been used to detect human activities (Koppula, Gupta, and Saxena 2013; Ni, Wang, and Moulin 2013; Sung et al. 2012) and provide rich information including human skeleton movement data and 3D environment maps. Improved methods for reasoning about human pose and intentions are therefore needed. Critical to this task are the roles of high-dimensionality and uncertainty. Many co-robotics tasks are

performed within high-dimensional control spaces where there are a variety of ways for a human to reasonably accomplish the task. Behavior modelling techniques for intent recognition and trajectory forecasting must scale to these high-dimensional control spaces and incorporate uncertainty over inherently ambiguous human behavior.

We present an inverse optimal control (IOC) approach using Linear-Quadratic Regulation (LQR) for intention recognition and trajectory forecasting of tasks involving hand motion trajectories. We apply the recently developed technique of maximum entropy IOC for LQR (Ziebart, Bagnell, and Dey 2010; Ziebart, Dey, and Bagnell 2012; Levine and Koltun 2012), allowing us to efficiently scale the inverse optimal control approach to continuous-valued three-dimensional positions, velocities, accelerations, etc.. Our formulation is inherently probabilistic and enables the inference of the user’s intent based on partial behavior observations and forecasts of future behavior.

## Related Work

There has been a significant amount of recent work on forecasting the future behavior of people to improve intelligent systems. In the robotic navigation domain, this has been manifested in robots that plan paths that are complementary to a pedestrian’s future movements (Ziebart et al. 2009b) or navigate through crowds based on anticipated movements (Henry et al. 2010; Trautman and Krause 2010; Kuderer et al. 2012). In robotic manipulation, techniques that interpret and aid in realizing a teleoperator’s intentions to complete a task (Hauser 2013) have had success.

Our work is most closely related, but complementary, to anticipatory temporal conditional random fields (ATCRF) (Koppula and Saxena 2013). Under that approach, discriminative learning is employed to model the relationships between object affordances and sub-activities at the “discrete” level and a simple generative model (based on a Gaussian distribution) of human pose and object location trajectories is employed at the “continuous” level.

We extend discriminative learning techniques, in the form of inverse optimal control, to the continuous level of human pose trajectories. The two approaches are complementary in that any inferred object affordances and sub-activities at the discrete level can be employed to shape the prior distributions at the continuous level, and the posterior inferences at

the continuous-level can feed into inferences at the discrete level. Our approach differs in its applicability to continuous-valued control settings and in employing a maximum entropy formulation rather than a non-probabilistic maximum-margin estimation approach.

Early work was conducted in inverse optimal control (Ng and Russell 2000) to learn appropriate control policies for planning and navigation (Ratliff, Silver, and Bagnell 2009). These, and related maximum margin approaches (Ratliff, Bagnell, and Zinkevich 2006), are non-probabilistic. Recent probabilistic approaches (Ziebart et al. 2009b; Wang et al. 2012) construct probability distributions over behaviors and can be used to infer unknown intentions or future behavior given a partially-completed observation. Maximum entropy inverse optimal control (Ziebart et al. 2008) has been shown to elicit strong results by maximizing the uncertainty in the probabilistic inference task (Ziebart et al. 2009a). Extensions to the linear-quadratic setting have been applied to predicting the intended targets of computer cursor movements (Ziebart, Dey, and Bagnell 2012) detailing the benefit of using the demonstrated continuous trajectory for target prediction. There has also been recent work in the area of action inference and activity forecasting from video (Kitani et al. 2012), however this work discretizes the state space in such a way that would be impractical for inferring at useful granularities in higher dimensional (three or more) spaces.

We extend the maximum entropy inverse optimal control linear quadratic regulator model (Ziebart, Dey, and Bagnell 2012) to the task of predicting target intentions and inferring continuous hand motion trajectories using depth camera data. This is directly applicable to the areas of activity prediction (Koppula and Saxena 2013) and behavior forecasting (Ziebart et al. 2009b).

## Approach

We propose a predictive model of motion trajectories trained from a dataset of observed depth camera motions using a linear quadratic regulation framework. Under this framework, a linear state dynamics model and a quadratic cost function are assumed. The optimal control problem for this formulation is to find the best control action for each possible state that minimizes the expectation of the cost function over time. In contrast, we estimate the cost function that best rationalizes demonstrated trajectories in this work. Our approach provides a probabilistic model that allows us to reason about the intentions of partially completed trajectories.

### State Representation and Quadratic Cost Matrices

One of the main benefits of the proposed technique is the ability to use a simple and generalizable state model. For the task at hand, we consider the  $xyz$  positions of the center of the hand as well as the velocities and accelerations in each respective axis. Adding one constant term to capture the linear cost, we have 10 dimensions in the state representation. The state at time step  $t$  is then defined as:

$$\mathbf{s}_t = [x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \ddot{x}_t, \ddot{y}_t, \ddot{z}_t, 1]^T,$$

where  $(\dot{x}_t, \dot{y}_t, \dot{z}_t)$  represent the velocities, and  $(\ddot{x}_t, \ddot{y}_t, \ddot{z}_t)$  represent the accelerations. A constant of 1 is added to the state representation in order to incorporate linear features into the quadratic cost function formulation described later.

Importantly, when actions,  $\mathbf{a}_t$  represent the velocities specifying changes in positions between current state and next state, the state dynamics follow a linear relationship:  $\mathbf{s}_{t+1} = \mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{a}_t + \epsilon_t$ , where the noise,  $\epsilon_t$ , is drawn from a zero-mean Gaussian distribution. We represent the distribution over next state using the transition probability distribution  $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ . Though many tasks may have non-linear dynamics or be motivated by non-quadratic cost functions, the LQR simplification is important when considering the fast execution time needed for true human-robot interaction. If needed, additional features can be added to the state in order to further incorporate higher-order attributes of the trajectory, like jerks. In addition to efficiency, this state definition provides a strong set of informative features for defining (or estimating) a cost function. While the position is clearly correlated with the intended target, the motion dynamics offer a more distinct descriptive behavior in terms of completing differing tasks (Filipovych and Ribeiro 2007).

In order to account for the high level of variance with respect to the  $xyz$  goal positions of the demonstrated trajectories obtained from the depth camera data, we extend the LQR method (Ziebart, Dey, and Bagnell 2012) and add an additional parameter matrix to the existing cost matrix  $\mathbf{M}$ . We call this matrix  $\mathbf{M}_f$  to signify that it only applies to the final state of the trajectory. This set of parameters adds an additional penalty to the final state for deviating far from the desired goal state features. So the cost functions are:

$$\text{cost}(\mathbf{s}_t, \mathbf{a}_t) = \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \mathbf{M} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}, t < T, \quad (1)$$

$$\text{cost}(\mathbf{s}_T) = (\mathbf{s}_T - \mathbf{s}_G)^T \mathbf{M}_f (\mathbf{s}_T - \mathbf{s}_G). \quad (2)$$

To increase the generalization of the model to all possible  $xyz$  conditions, we sparsify the two parameter matrices so that we only train parameters relating to the dynamics features, velocity and acceleration, for  $\mathbf{M}$ , and only train parameters relating to  $xyz$  positions for  $\mathbf{M}_f$ . Therefore, all quadratic feature combinations that include an  $xyz$  position feature in  $\mathbf{M}$  or a velocity, or acceleration, feature in  $\mathbf{M}_f$  will have a parameter value of 0. This is helpful in the case where the  $xyz$  coordinates differ greatly across the demonstrated trajectories and allows for  $\mathbf{M}$  to learn the behavior of the trajectories rather than the true positions. Likewise,  $\mathbf{M}_f$  can then serve solely as a position penalty for the final inferred position deviating from the desired goal location and avoid penalizing the velocity and acceleration of the final state. This helps to generalize the model across different orientations and behaviors.

### Inverse Linear-Quadratic Regulation for Prediction

We employ maximum causal entropy inverse optimal control (Ziebart, Bagnell, and Dey 2013) to the linear quadratic regulation setting to obtain estimates for our cost parameter

matrices,  $\mathbf{M}$  and  $\mathbf{M}_f$ . These estimates result from a constrained optimization problem maximizing causal entropy,

$$H(\mathbf{a}|\mathbf{s}) \triangleq \mathbb{E}_{\hat{\pi}} \left[ - \sum_{t=1}^T \log \hat{\pi}(\mathbf{a}|\mathbf{s}) \right],$$

so that the predictive policy distribution,  $\hat{\pi}(\mathbf{a}|\mathbf{s}) = \pi(\mathbf{a}_1|\mathbf{s}_1)\pi(\mathbf{a}_2|\mathbf{s}_2) \cdots \pi(\mathbf{a}_T|\mathbf{s}_T)$ , matches the quadratic state properties of the demonstrated behavior,  $\tilde{\pi}$ , in feature expectation where:

$$\mathbb{E}_{\hat{\pi}} \left[ \sum_{t=1}^{T-1} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \right] = \mathbb{E}_{\tilde{\pi}} \left[ \sum_{t=1}^{T-1} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \right] \text{ and } \quad (3)$$

$$\mathbb{E}_{\hat{\pi}}[(\mathbf{s}_T - \mathbf{s}_G)(\mathbf{s}_T - \mathbf{s}_G)^T] = \mathbb{E}_{\tilde{\pi}}[(\mathbf{s}_T - \mathbf{s}_G)(\mathbf{s}_T - \mathbf{s}_G)^T].$$

This set of constraints ensures that the trajectory’s dynamic properties are maintained by the control policy estimate,  $\hat{\pi}$ . Solving this problem, we obtain a state-conditioned probabilistic policy  $\hat{\pi}$  over actions that are recursively defined using the following equations:

$$\hat{\pi}(\mathbf{a}_t|\mathbf{s}_t) = e^{Q(\mathbf{s}_t, \mathbf{a}_t) - V(\mathbf{s}_t)}, \quad (4)$$

$$Q(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\tau(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)} [V(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)] + \text{cost}(\mathbf{s}_t, \mathbf{a}_t), \quad (5)$$

$$V(\mathbf{s}_t) = \begin{cases} \text{softmax}_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t), & t < T \\ (\mathbf{s}_t - \mathbf{s}_G)^T \mathbf{M}_f (\mathbf{s}_t - \mathbf{s}_G), & t = T, \end{cases} \quad (6)$$

where  $\mathbf{s}_G$  represents a goal state that is used to specify a penalty for a trajectory’s final distance from the desired goal and the softmax function is a smoothed interpolation of the maximum function,  $\text{softmax}_x f(x) = \log \int_x e^{f(x)} dx$ .

Using this formulation, the recurrence of Equation 5 and 6 can be viewed as a probabilistic relaxation of the Bellman criteria for optimal control; it selects actions with smaller expected future costs in Equation 5 and recursively computes those future costs using the expectation over the decision process’s dynamics in Equation 6. The  $Q$  and  $V$  functions are softened versions of the optimal control state-action and state value functions, respectively. Inference is performed by forming a set of time-specific update rules that are used to compute state and action values which are conditioned on the goal state, trajectory length, and the cost parameter values. We refer to the appendix for details of this method. The probability of a trajectory of length  $T$  is then easily

$$\text{computed as: } \prod_{\tau=1}^t \pi(a_\tau|\mathbf{s}_\tau, G, I, T) = e^{\sum_{\tau=1}^{t-1} Q(a_\tau, \mathbf{s}_\tau) - V(\mathbf{s}_t)},$$

where  $G$  is the goal location of the trajectory and  $I$  is the characteristic intention/behavior of the movement (ex. reaching, placing, eating, etc.).

The recursive  $Q$  and  $V$  calculations simplify to a set of matrix updates in the LQR setting:

$$Q(\mathbf{s}_t, \mathbf{a}_t) = \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_{a_t, a_t} & \mathbf{C}_{a_t, s_t} \\ \mathbf{C}_{s_t, a_t} & \mathbf{C}_{s_t, s_t} \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} + \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{F}_{a_t} \\ \mathbf{F}_{s_t} \end{bmatrix} + Q_{cv_t},$$

$$V(\mathbf{s}_t) = \mathbf{s}_t^T \mathbf{D}_t \mathbf{s}_t + \mathbf{s}_t^T \mathbf{G}_t + V_{cv_t},$$

where  $Q_{cv_t}$  and  $V_{cv_t}$  are scalars. The matrices of these values are recursively defined as:

$$\mathbf{D}_T = \mathbf{M}_f; \quad \mathbf{G}_T = -2\mathbf{M}_f \mathbf{s}_G;$$

for  $t$  in  $T - 1 \dots 1$

$$\mathbf{C}_{a_t, a_t} = \mathbf{B}^T \mathbf{D}_{t+1} \mathbf{B} + \mathbf{M}_{a, a}; \quad \mathbf{C}_{a_t, s_t} = \mathbf{B}^T \mathbf{D}_{t+1} \mathbf{A} + \mathbf{M}_{a, s};$$

$$\mathbf{C}_{s_t, a_t} = \mathbf{A}^T \mathbf{D}_{t+1} \mathbf{B} + \mathbf{M}_{s, a}; \quad \mathbf{C}_{s_t, s_t} = \mathbf{A}^T \mathbf{D}_{t+1} \mathbf{A} + \mathbf{M}_{s, s};$$

$$\mathbf{F}_{a_t} = \mathbf{B}^T \mathbf{G}_{t+1}; \quad \mathbf{F}_{s_t} = \mathbf{A}^T \mathbf{G}_{t+1};$$

$$\mathbf{D}_t = \mathbf{C}_{s_{t+1}, s_{t+1}} - \mathbf{C}_{a_{t+1}, s_{t+1}}^T \mathbf{C}_{a_{t+1}, a_{t+1}}^{-1} \mathbf{C}_{a_{t+1}, s_{t+1}};$$

$$\mathbf{G}_t = \mathbf{F}_{s_{t+1}} - \mathbf{C}_{a_{t+1}, s_{t+1}}^T \mathbf{C}_{a_{t+1}, a_{t+1}}^{-1} \mathbf{F}_{a_{t+1}}.$$

We derive these updates in a supplemental appendix.

## Bayesian Intention and Target Prediction

Given a unique learned trajectory model for each intention,  $I$ , the observed partial state sequence,  $s_{1:t}$ , the prior probability distribution over the duration of the total sequence,  $T$ , and the goal,  $G$ , we employ Bayes’ rule to form the posterior probability distribution of the possible targets:

$$P(G, I, T | s_{1:t}) \propto \prod_{\tau=1}^t \pi(a_\tau | s_\tau, G, I, T) P(G, I, T). \quad (7)$$

The Bayesian formulation enables us to use a flexible prior distribution over the goals and intentions that reflects our initial belief given relevant environment information, such as pre-defined task type, object affordance and distance with objects. This allows for any method that estimates a strong prior distribution to be combined with the LQR model in order to improve the predictive accuracy.

## Complexity Analysis

When the dynamics are linear, the recursive relation of Equations 5 and 6 have closed-form solutions that are quadratic functions and the action distribution is a conditional Gaussian distribution. This enables more efficient computation for large time horizons,  $T$ , with a large continuous state and action space than could be computed in the discrete case.

For trajectory inference, the proposed technique only requires  $O(T)$  matrix updates as detailed in the appendix. A strong advantage of this model is the fact that the matrix updates only need to be computed once when performing inference over sequences sharing the same time horizon and goal position. In order to further improve the efficiency of our computation we employ the Armadillo C++ linear algebra library for fast linear computation (Sanderson 2010).

## Experimental Setup

We employ the Cornell Activity Dataset (CAD-120) (Kopula and Saxena 2013) in order to analyze and evaluate our predictive inverse linear-quadratic regulation model.

### Cornell Activity Dataset

The dataset consists of 120 depth camera videos of long daily activities. There are 10 high-level activities: making cereal, taking medicine, stacking objects, unstacking objects, microwaving food, picking objects, cleaning objects,

taking food, arranging objects and having a meal. These high-level activities are then divided into 10 sub-activities: reaching, moving, pouring, eating, drinking, opening, placing, closing, cleaning and null. Here, we regard different types of sub-activity as different intentions.

Consider the task of pouring cereal into a bowl. We decompose this high-level activity into the following sub-activities: reaching (cereal box), moving (cereal box above bowl), pouring (from cereal box to a bowl), moving (cereal box to table surface), placing (cereal box to table surface), null (moving hand back).

**Modifications** The goal of the moving sub-activity is dependent on the sub-activity it precedes. For instance, if the next sub-activity is placing, then the goal of moving will be some location above the target surface of the placing sub-activity, but if the next sub-activity is eating, then the goal will be the area around the mouth. Therefore, we regard the moving sub-activity as the beginning portion of the latter sub-activity and combine them as one. We ignore the null sub-activity due to its lack of intention or goal.

In a similar fashion, we also separate the opening sub-activity into two different sub-activities, opening a microwave and opening a jar. The actions in these two tasks have very different movements and goals, and thus are considered as separate sub-activities. This results in nine sub-activity classifications, each of which we regard as a trajectory type with a target and intent.

**Test set** The data is randomly divided into a training set and a test set. The test set consists of 10% of the demonstrated sequences with at least one sequence belonging to each sub-activity. The model is then trained on the training set and evaluated using the test set.

## Model Fitting

**Estimating the Quadratic Parameters** Our LQR model uses two separate parameter matrices,  $M$  and  $M_f$ . We employ accelerated stochastic gradient descent with an adaptive (adagrad) learning rate and  $L_1$  regularization (Duchi, Hazan, and Singer 2011; Sutskever et al. 2013) on both parameter matrices simultaneously. This regularized approach prevents overfitting of the parameter matrices for sub-activities with a low number of demonstrated trajectories.

## Target and Intention Sampling

**Intention Sampling** In the prediction task we separate the sub-activities into two categories, those that require an object in the hand and those that do not. This eliminates tasks from consideration that lack the required presence or absence of an object. For example, when the task is placing an object and there is no object in the hand.

**Target Sampling** Target points are chosen for each sub-activity according to a Gaussian distribution of the observed endpoints of the active hand and the target object in the training data. The target object for the eating and drinking sub-activities are the head joint and the target for the placing sub-activity is the true endpoint of the active hand trajectory. The skeleton and object tracking data are then used with this

observed distribution to compute the probable endpoints for the test trajectories. Since the placing sub-activity has no target object, points are randomly chosen on the placeable surfaces in the scene.

**Segmentation and Duration Sampling** When deployed in real world applications the separation of intentions may not be clear. For this reason it is necessary to include a method to detect segmentation. The incorporation of  $T$  in Equation 7 allows for the direct inclusion of segmentation inference as an inherent feature of the proposed LQR method. This is due to the fact that goals, intentions, and duration are grouped together in each prediction.

The inclusion of duration is important as a poor prediction of the end of the current sub-activity will lead to an inaccurate starting step for the next sub-activity. For this reason it is necessary to achieve an accurate estimate of the length of each sub-activity.

Our LQR model requires that we specify a length for the total trajectory. Since we are only observing a part of this trajectory prior to forming a prediction, it is necessary to infer this duration. Here we use the observed average distance covered by each action for each sub-activity with the remaining distance to be covered from the current point and the sampled target point in order to infer a probable duration. This is done using the following equation:  $T = t + \text{dist}(s_t, s_g) / \text{avgdist}_a$ . This value,  $T$ , represents the inferred segmentation for sub-activity  $a$  and target  $s_g$ .

## Prior Distributions

In order to improve the predictive accuracy, it is often helpful to include a prior distribution on the sampled target points. This allows for the inclusion of additional knowledge about the problem domain to be added to the inference model. Here, the priors enable us to use the context information of the human robot interaction which is not included in trajectory data in our model. We use two prior distributions for this purpose which we join together into a single distribution due to the tuple nature of the sampled sub-activity and target points.

We evaluate the inverse LQR technique against 4 different probabilistic prediction methods, nearest target, nearest target from extended trajectory, a sub-activity sequence Naive Bayes distribution, and a combination of nearest target and the sub-activity sequence distribution which is also used as a prior distribution for our proposed LQR method.

**Target Distance Prior** For the nearest target prior we use the Euclidean distance from the final state of the observed trajectory,  $s_t$ , and the sampled target,  $s_g$ , and a strength coefficient,  $\alpha$ , in order to form a probability distribution over the possible target points using the following equation:

$$p(s_g | s_t) \propto e^{-\alpha \text{dist}(s_t, s_g)}. \quad (8)$$

**Markov Intention Prior** Here we use a second order Markov model to form the probability distribution of a sub-activity given the previous two observed sub-activities in the sequence:

$$p(a_i | a_{i-1}, a_{i-2}) = \frac{N(a_i, a_{i-1}, a_{i-2})}{N(a_{i-1}, a_{i-2})}, \quad (9)$$

with  $N(a_i, a_{i-1}, a_{i-2})$  being the number of occurrences of activity  $a_i$  being preceded by the two activities,  $a_{i-1}$  and  $a_{i-2}$  in the training dataset.

**Combining for a Full Prior** The above two prior distributions are then combined together in order to form a distribution over both the sub-activity and the location of the sampled target points with a simple method:  $p(s_g, a_i) \propto p(a_i|a_{i-1}, a_{i-2})p(s_g|s_t)$ .

## Prediction

**Current and Next Target** We calculate probabilistic predictions for each sampled target using our LQR method (7). Here the trajectory of the most active hand is observed and used to develop a likelihood model for each target.

When predicting the probability of the next target (the target following the not-yet-reached current target), we use the calculated probability of the current target and the target points in order to develop a distribution for the next target point. Since there is no trajectory to observe for the next target, this reduces to simply using the prior distribution of the next targets and the probability distribution of the current target given the current partial trajectory:  $P(G_{i+1}, I_{i+1}, T_{i+1}|s_{1:t}) \propto P(G_i, I_i, T_i|s_{1:t})P(G_{i+1}, I_{i+1}, T_{i+1}|G_i, I_i, T_i)$ .

**Notes on Segmentation Prediction** In addition to the direct inference of segmentation, many sub-activities contain clear discernible goal behaviors. For instance, the placing sub-activity will end with the active hand releasing an object. If the object is tracked, which is necessary for target sampling, we can easily determine when this takes place and the hand moves away from the placed object. This is also the case for reaching, which ends when the hand is in contact with an object.

Utilizing these characteristics and accurate estimates of the duration yields impressive results for segmentation inference. We are able to achieve perfect segmentation detection once the hands move close enough to the inferred target to get an accurate estimate of the remaining duration, notably once the goal is reached or passed. This is also evident in the improved log-loss, which represents the probability of the true duration, goal, and intention.

## Evaluation

### Comparison Metrics

We evaluate the inverse LQR technique against the probability distributions obtained from calculating the three prior methods. These are the nearest target distribution, the Markov model on the sub-activity sequence, and the full prior distribution that is a combination of the two.

We compare these distributions against the LQR model with a flat distribution prior, and with each of the above respective distributions as priors.

### Execution Time

In many tasks using predictive inference, it is important to minimize the execution time of the inference task. In this case, the proposed LQR method models, infers and makes

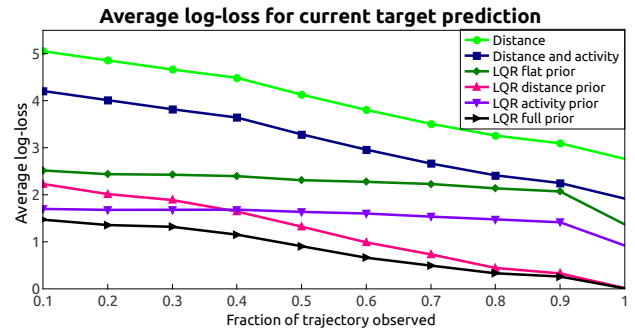


Figure 1: Average predictive loss of current target with partially observed trajectories. We compare the results of the distance and the distance-activity full prior with LQR.

predictions for the 63 trajectories of the test set in 1.052 seconds total with the parallel training of the parameters for all nine of the sub-activity types taking less than an hour. These execution times were collected on an Intel i7-3720QM CPU at 2.60GHz with 16 GB of RAM.

The use of linear dynamics in the models allows for this efficient computation which is extremely important when inferring over quickly executed sequences that require fast reactions using the predicted results. In addition, relative positions were not used in the state formulation since the time to compute a different relative sequence of states/actions for each possible target combination could result in non-real-time computation which is of key importance for human-robot-interaction. The speeds described above for the entire dataset translates to a target inference rate of over 1000Hz for real-time applications. This is a frequency that is many orders of magnitude greater than the discrete ATCRF method of inference (Koppula and Saxena 2013).

## Predictive Results

We compare the predictive accuracy of our method against the aforementioned techniques using the averaged mean log-loss (Begleiter, El-yaniv, and Yona 2004; Nguyen and Guo 2007). This allows us to compare the likelihood of the demonstrated trajectory to the distance and activity measures previously discussed. As we show in Figure 1, the presented LQR method outperforms the other predictive techniques. This is mainly due to the incorporation of our sophisticated LQR likelihood model for the demonstrated sequence trajectories with the prior target distribution.

The activity prior is independent of the percentage of the sequence that is observed with an averaged mean log-loss for the activity model of 4.54. While this is not very good on its own, it does add a significant improvement to the LQR model when incorporated as a prior. Likewise, the nearest-target distance model gives an additional boost to the results of the LQR model when added as a prior with the combination of the two priors gave the best results. The improvement seen in Figure 1 is then extended to the predictive results for the target of the next sub-activity in Figure 2. Each of the models uses the same technique to compute the next target

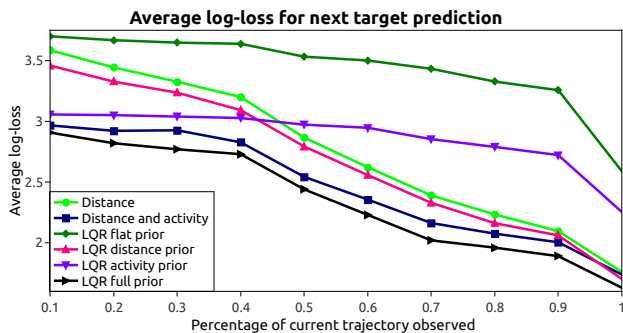


Figure 2: Average predictive loss of next target with partially observed trajectories. We compare the results of the distance and the distance-activity full prior with LQR.

Results Comparison with Ground Truth Segmentation			
Method	Accuracy	Macro Precision	Macro Recall
LQR 20% sequence	80.9 ± 2.4	65.0 ± 3.1	77.3 ± 2.4
LQR 40% sequence	82.5 ± 3.2	73.4 ± 2.2	91.4 ± 0.6
LQR 60% sequence	84.1 ± 0.9	79.1 ± 2.5	94.2 ± 0.6
LQR 80% sequence	90.4 ± 0.4	87.5 ± 1.8	96.2 ± 0.3
LQR 100% sequence	100 ± 0.0	100 ± 0.0	100 ± 0.0
ATCRF 100% sequence	86.0 ± 0.9	84.2 ± 1.3	76.9 ± 2.6

Results Comparison without Ground Truth Segmentation			
Method	Accuracy	Macro Precision	Macro Recall
LQR 20% sequence	66.7 ± 3.9	50.1 ± 3.7	62.4 ± 3.8
LQR 40% sequence	69.8 ± 3.9	50.5 ± 4.4	56.7 ± 3.3
LQR 60% sequence	76.1 ± 2.7	72.2 ± 2.7	93.3 ± 0.5
LQR 80% sequence	77.8 ± 3.4	75.7 ± 2.9	93.5 ± 0.5
LQR 100% sequence	100 ± 0.0	100 ± 0.0	100 ± 0.0
ATCRF 100% sequence	68.2 ± 0.3	71.1 ± 1.9	62.2 ± 4.1

Table 1: Accuracy and macro precision and recall with standard error for current activity detection when different percentages of the sequence are observed.

distribution and then their respective methods for the computation of the current target distribution.

While the focus of this work is to improve the predictive log-loss of the model, it is helpful to use a classification accuracy in order to allow us to compare our results to prior techniques used on this data. In this case we choose the most probable target given the partially observed trajectory as the classified goal. Table 1 compares the results of the presented LQR technique with the ATCRF method (Koppula and Saxena 2013) where macro precision and recall are the averages of precision and recall respectively for all sub-activities. As we show in Table 1, our LQR method has significantly improved upon the previous state of the art results with perfect classification accuracy when detecting the activity given the entire demonstrated trajectory. This is especially true when ground truth segmentation is not given. A likely reason for this is that obtaining perfect segmentation detection allows

for us to use the true starting point for each sub-activity.

In addition, the LQR technique, given only 40% of the sequence obtains comparable results to the ATCRF model given the entire sequence. This is a significant result since the ability to predict the target and intention early on in a sequence is highly beneficial to many applications including human-robot interaction.

We note that the ATCRF results are on the unmodified CAD-120 dataset. As mentioned previously, we have merged the moving sub-activity with its succeeding sub-activity and separated the opening task into two different sub-activities. While this produces a more straight forward sampling task, it also makes the first half of the demonstrated sequences more ambiguous and segmentation more difficult. This is due to the similar dynamics of the moving sub-activity as compared to the other sub-activities. However, our results show that the proposed LQR method is robust enough to generate strong performance after only observing the first 20% of the sequence.

## Discussion

In this paper we have shown that incorporating the dynamics of a trajectory sequence into a predictive model elicits a significant improvement in the inference of target locations and activities for human task completion. We did this using linear quadratic regulation trained with maximum entropy inverse optimal control and have shown that using linear dynamics in forming a model for task and target prediction improves intention recognition while providing efficient computation of the inferred probability distributions.

The combination of efficient inference with strong predictive results yields a very promising technique for any field that requires the predictive modelling of decision processes in real time. This is especially important in the area of human-robot collaboration where a robot may need to react to the inferred intentions of a human collaborator before they complete a task, which is an application that the authors plan to undertake in the near future.

While the results reported improve upon the conditional random field (ATCRF) technique (Koppula and Saxena 2013), we feel the best results can be obtained by incorporating the discrete distribution learned using the ATCRF as an additional prior into the proposed LQR model. Any methods that improve the prior distribution (e.g., (Sung et al. 2012; Koppula and Saxena 2013)) will improve the log-loss since it is additive over the likelihood function and prior when taking the log of Equation 7. This combination of discrete and continuous learned models should return a strong predictive distribution of the targets and intentions by accounting for both the linear dynamics of the motion trajectories and the object affordances of the activity space during inference.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. #1227495.

## References

- Begleiter, R.; El-yaniv, R.; and Yona, G. 2004. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*.
- Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121–2159.
- Filipovych, R., and Ribeiro, E. 2007. Combining models of pose and dynamics for human motion recognition. In *International Symposium on Visual Computing*.
- Hauser, K. 2013. Recognition, prediction, and planning for assisted teleoperation of freeform tasks. *Autonomous Robots*.
- Henry, P.; Vollmer, C.; Ferris, B.; and Fox, D. 2010. Learning to navigate through crowded environments. In *International Conference on Robotic Automation*.
- Kidokoro, H.; Kanda, T.; Brscic, D.; and Shiomi, M. 2013. Will I bother here? A robot anticipating its influence on pedestrian walking comfort. In *Human Robot Interaction*.
- Kitani, K.; Ziebart, B.; Bagnell, J.; and Hebert, M. 2012. Activity forecasting. In *Computer Vision European Conference on Computer Vision 2012*.
- Koppula, H., and Saxena, A. 2013. Anticipating human activities using object affordances for reactive robotic response. In *Robotics: Science and Systems*.
- Koppula, H. S.; Gupta, R.; and Saxena, A. 2013. Learning human activities and object affordances from rgb-d videos. *International Journal on Robotic Research*.
- Kuderer, M.; Kretzschmar, H.; Sprunk, C. R. I.; and Burgard, W. 2012. Feature-based prediction of trajectories for socially compliant navigation. In *Robotics: Science and Systems*.
- Levine, S., and Koltun, V. 2012. Continuous inverse optimal control with locally optimal examples. In *International Conference on Machine Learning*.
- Ng, A. Y., and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, 663–670. Morgan Kaufmann.
- Nguyen, N., and Guo, Y. 2007. Comparisons of sequence labeling algorithms and extensions. In *International Conference on Machine Learning*.
- Ni, B.; Wang, G.; and Moulin, P. 2013. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *Consumer Depth Cameras for Computer Vision*.
- Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; and Thrun, S. 2003. Towards robotic assistants in nursing homes: Challenges and results. *Special issue on Socially Interactive Robots, Robotics and Autonomous Systems*.
- Ratliff, N. D.; Bagnell, J. A.; and Zinkevich, M. A. 2006. Maximum margin planning. In *International Conference on Machine Learning*.
- Ratliff, N. D.; Silver, D.; and Bagnell, J. A. 2009. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*.
- Sanderson, C. 2010. Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments. Technical report, NICTA.
- Strabala, K.; Lee, M. K.; Dragan, A.; Forlizzi, J.; Srinivasa, S.; Cakmak, M.; and Micelli, V. 2013. Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*.
- Sung, J.; Ponce, C.; Selman, B.; and Saxena, A. 2012. Unstructured human activity detection from RGBD images. In *International Conference on Robotic Automation*.
- Sutskever, I.; Martens, J.; Dahl, G. E.; and Hinton, G. E. 2013. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, 1139–1147.
- Trafton, J. G.; Hiatt, L. M.; Harrison, A. M.; Tamborello, P.; Khemlani, S. S.; and Schultz, A. C. 2013. Act-r/e: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*.
- Trautman, P., and Krause, A. 2010. Unfreezing the robot: Navigation in dense, interacting crowds. In *International Conference on Intelligent Robots and Systems*.
- Wang, Z.; Deisenroth, M. P.; Amor, H. B.; Vogt, D.; Schlkopf, B.; and Peters, J. 2012. Probabilistic modeling of human movements for intention inference. In *Robotics: Science and Systems*.
- Ziebart, B. D.; Bagnell, J. A.; and Dey, A. K. 2010. Modeling interaction via the principle of maximum causal entropy. In *International Conference on Machine Learning*.
- Ziebart, B. D.; Bagnell, J. A. D.; and Dey, A. 2013. The principle of maximum causal entropy for estimating interacting processes. *IEEE Transactions on Information Theory*.
- Ziebart, B. D.; Maas, A.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Association for the Advancement of Artificial Intelligence*.
- Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2009a. Human behavior modeling with maximum entropy inverse optimal control. In *Association for the Advancement of Artificial Intelligence Spring Symposium: Human Behavior Modeling*.
- Ziebart, B. D.; Ratliff, N.; Gallagher, G.; Mertz, C. R. I.; Peterson, K.; Bagnell, J. A.; Hebert, M.; Dey, A. K.; and Srinivasa, S. 2009b. Planning-based prediction for pedestrians. In *International Conference on Intelligent Robots and Systems*.
- Ziebart, B. D.; Dey, A. K.; and Bagnell, J. A. 2012. Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of the ACM International Conference on Intelligent User Interfaces*.

## Update rule derivation

Since recurrence values in Equations 5 and 6 are of quadratic forms, we break down the matrices into time dependent components:

$$Q(\mathbf{s}_t, \mathbf{a}_t) = \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_{a_t, a_t} & \mathbf{C}_{a_t, s_t} \\ \mathbf{C}_{s_t, a_t} & \mathbf{C}_{s_t, s_t} \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} + \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{F}_{a_t} \\ \mathbf{F}_{s_t} \end{bmatrix} + Q_{cv_t},$$

$$V(\mathbf{s}_t) = \mathbf{s}_t^T \mathbf{D}_t \mathbf{s}_t + \mathbf{s}_t^T \mathbf{G}_t + V_{cv_t},$$

where  $Q_{cv_t}$  and  $V_{cv_t}$  are the respective constants derived from the  $Q$  and  $V$  functions at time  $t$ .

Combining the above definitions with Equations 5 and 6 yields:

$$\begin{aligned} Q(\mathbf{s}_t, \mathbf{a}_t) &= \mathbb{E}_{\tau(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)} [\mathbf{s}_{t+1}^T \mathbf{D}_t \mathbf{s}_{t+1} + \mathbf{s}_{t+1}^T \mathbf{G}_t + V_{cv_t} | \mathbf{s}_t, \mathbf{a}_t] \\ &+ \text{cost}(\mathbf{s}_t, \mathbf{a}_t), \\ &= (\mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{a}_t)^T \mathbf{D}_t (\mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{a}_t) + \text{tr}(\mathbf{D}_t \Sigma) \\ &+ \mathbf{a}_t^T \mathbf{B}^T \mathbf{G}_t + \mathbf{s}_t^T \mathbf{A}^T \mathbf{G}_t + \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \mathbf{M} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{B}^T \mathbf{D}_t \mathbf{B} & \mathbf{B}^T \mathbf{D}_t \mathbf{A} \\ \mathbf{A}^T \mathbf{D}_t \mathbf{B} & \mathbf{A}^T \mathbf{D}_t \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{M}_{a_t, a_t} & \mathbf{M}_{a_t, s_t} \\ \mathbf{M}_{s_t, a_t} & \mathbf{M}_{s_t, s_t} \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} + \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{B}^T \mathbf{G}_t \\ \mathbf{A}^T \mathbf{G}_t \end{bmatrix}. \end{aligned}$$

When  $t < T$ :

$$\begin{aligned} V(\mathbf{s}_t) &= \ln \int_{\mathbf{a}_t} e^{\mathbf{a}_t^T \mathbf{C}_{a_t, a_t} \mathbf{a}_t + 2\mathbf{a}_t^T \mathbf{C}_{a_t, s_t} \mathbf{s}_t + \mathbf{s}_t^T \mathbf{C}_{s_t, s_t} \mathbf{s}_t + \mathbf{s}_t^T \mathbf{F}_{s_t} + \mathbf{a}_t^T \mathbf{F}_{a_t} + Q_{cv_t}} d\mathbf{a}_t \\ &= \mathbf{s}_t^T \mathbf{C}_{s_t, s_t} \mathbf{s}_t + \mathbf{s}_t^T \mathbf{F}_{s_t} + Q_{cv_t} \\ &+ \ln \left( \frac{1}{Z_t} \int_{\mathbf{a}_t} Z_t e^{-0.5 \mathbf{a}_t^T (-2\mathbf{C}_{a_t, a_t}) \mathbf{a}_t + \mathbf{a}_t^T (2\mathbf{C}_{a_t, s_t} \mathbf{s}_t + \mathbf{F}_{a_t})} d\mathbf{a}_t \right) \\ &= \mathbf{s}_t^T \mathbf{C}_{s_t, s_t} \mathbf{s}_t + \mathbf{s}_t^T \mathbf{F}_{s_t} + Q_{cv_t} \\ &+ \ln \left( \frac{1}{Z_t} \int_{\mathbf{a}_t} N[\mathbf{a}_t | 2\mathbf{C}_{a_t, s_t} \mathbf{s}_t + \mathbf{F}_{a_t}, -2\mathbf{C}_{a_t, a_t}] d\mathbf{a}_t \right) \\ &= \mathbf{s}_t^T \mathbf{C}_{s_t, s_t} \mathbf{s}_t + \mathbf{s}_t^T \mathbf{F}_{s_t} + Q_{cv_t} - \ln Z_t, \end{aligned}$$

where

$$Z_t = \frac{e^{(-\frac{1}{2})(2\mathbf{C}_{a_t, s_t} \mathbf{s}_t + \mathbf{F}_{a_t})^T (-2\mathbf{C}_{a_t, a_t})^{-1} (2\mathbf{C}_{a_t, s_t} \mathbf{s}_t + \mathbf{F}_{a_t})}}{|2\pi(-2\mathbf{C}_{a_t, a_t})^{-1}|^{1/2}}.$$

$$\begin{aligned} V(\mathbf{s}_t) &= \mathbf{s}_t^T (\mathbf{C}_{s_t, s_t} - \mathbf{C}_{a_t, s_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{C}_{a_t, s_t}) \mathbf{s}_t \\ &+ \mathbf{s}_t^T (\mathbf{F}_{s_t} - \mathbf{C}_{a_t, s_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t}) \\ &+ Q_{cv_t} + \frac{1}{2} \ln |2\pi(-2\mathbf{C}_{a_t, a_t})^{-1}| - \frac{1}{4} \mathbf{F}_{a_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t} \\ &= \mathbf{s}_t^T (\mathbf{C}_{s_t, s_t} - \mathbf{C}_{a_t, s_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{C}_{a_t, s_t}) \mathbf{s}_t \\ &+ \mathbf{s}_t^T (\mathbf{F}_{s_t} - \mathbf{C}_{a_t, s_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t}) + Q_{cv_t} \\ &+ \frac{1}{2} \ln |2\pi(-2\mathbf{C}_{a_t, a_t})^{-1}| - \frac{1}{4} \mathbf{F}_{a_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t}. \end{aligned}$$

Thus,

$$\mathbf{D}_t = \mathbf{C}_{s_t, s_t} - \mathbf{C}_{a_t, s_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{C}_{a_t, s_t},$$

$$\mathbf{G}_t = \mathbf{F}_{s_t} - \mathbf{C}_{a_t, s_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t},$$

with,

$$V_{cv_t} = Q_{cv_t} + \frac{1}{2} \ln |2\pi(-2\mathbf{C}_{a_t, a_t})^{-1}| - \frac{1}{4} \mathbf{F}_{a_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t}.$$

The set of update rules for the quadratic functions are:

$$\mathbf{D}_T = \mathbf{M}_f;$$

$$\mathbf{G}_T = -2\mathbf{M}_f \mathbf{s}_G;$$

for  $t$  in  $T-1 \dots 1$

$$\mathbf{C}_{a_t, a_t} = \mathbf{B}^T \mathbf{D}_{t+1} \mathbf{B} + \mathbf{M}_{a_{t+1}, a_{t+1}};$$

$$\mathbf{C}_{a_t, s_t} = \mathbf{B}^T \mathbf{D}_{t+1} \mathbf{A} + \mathbf{M}_{a_{t+1}, s_{t+1}};$$

$$\mathbf{C}_{s_t, a_t} = \mathbf{A}^T \mathbf{D}_{t+1} \mathbf{B} + \mathbf{M}_{s_{t+1}, a_{t+1}};$$

$$\mathbf{C}_{s_t, s_t} = \mathbf{A}^T \mathbf{D}_{t+1} \mathbf{A} + \mathbf{M}_{s_{t+1}, s_{t+1}};$$

$$\mathbf{F}_{a_t} = \mathbf{B}^T \mathbf{G}_{t+1};$$

$$\mathbf{F}_{s_t} = \mathbf{A}^T \mathbf{G}_{t+1};$$

$$\mathbf{D}_t = \mathbf{C}_{s_{t+1}, s_{t+1}} - \mathbf{C}_{a_{t+1}, s_{t+1}}^T \mathbf{C}_{a_{t+1}, a_{t+1}}^{-1} \mathbf{C}_{a_{t+1}, s_{t+1}};$$

$$\mathbf{G}_t = \mathbf{F}_{s_{t+1}} - \mathbf{C}_{a_{t+1}, s_{t+1}}^T \mathbf{C}_{a_{t+1}, a_{t+1}}^{-1} \mathbf{F}_{a_{t+1}}.$$

The model parameters in matrix  $M$  and  $M_f$  are fit from the demonstrated examples. The likelihood of demonstration, is a convex function of those parameters. This guarantees that gradient-based techniques will converge to the value that best explains the demonstration. Under the maximum entropy framework, the gradients have intuitive interpretation: they are just the differences of the optimization constraints,

$$\nabla_M L = \mathbb{E}_{\hat{\pi}} \left[ \sum_{t=1}^T \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \right] - \mathbb{E}_{\hat{\pi}} \left[ \sum_{t=1}^T \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \right],$$

$$\nabla_{M_f} L = \mathbb{E}_{\hat{\pi}} [(s_T - s_G)(s_T - s_G)^T] - \mathbb{E}_{\hat{\pi}} [(s_T - s_G)(s_T - s_G)^T].$$

These values can be directly obtained from the Gaussian distribution of the state over time. Namely, if  $x$  is normally distributed with mean  $\mu_x$  and covariance matrix  $\Sigma_x$ ,  $\mathbb{E}[xx^T] = \mu_x \mu_x^T + \Sigma_x$ . Here, according to the Gaussian properties, if the distribution of  $s_t$  is  $N(\mu_{s_t}, \Sigma_{s_t})$ :

$$\begin{aligned} \log(\hat{\pi}(\mathbf{a}_t | \mathbf{s}_t)) &= Q(\mathbf{s}_t, \mathbf{a}_t) - V(\mathbf{s}_t) \\ &= \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_{a_t, a_t} & \mathbf{C}_{a_t, s_t} \\ \mathbf{C}_{s_t, a_t} & \mathbf{C}_{s_t, s_t} \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} + \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \begin{bmatrix} \mathbf{F}_{a_t} \\ \mathbf{F}_{s_t} \end{bmatrix} - \mathbf{s}_t^T \mathbf{D}_t \mathbf{s}_t - \mathbf{s}_t^T \mathbf{G}_t \\ &- \frac{1}{2} \log |2\pi(-2\mathbf{C}_{a_t, a_t})^{-1}| + \frac{1}{4} \mathbf{F}_{a_t}^T \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t}, \end{aligned}$$

which means  $\hat{\pi}(\mathbf{a}_t | \mathbf{s}_t) \propto N[\mathbf{a}_t | 2\mathbf{C}_{a_t, s_t} \mathbf{s}_t + \mathbf{F}_{a_t}, -2\mathbf{C}_{a_t, a_t}]$ ,

which is  $N((- \mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t} \mathbf{s}_t - \frac{1}{2} \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t}, (-2\mathbf{C}_{a_t, a_t})^{-1})$ .



So  $a_t \sim N(\mu_{a_t}, \Sigma_{a_t})$  where:

$$\begin{aligned}\mu_{a_t} &= -\mathbf{C}_{a_t, a_t}^{-1} \mathbf{C}_{a_t, s_t} \mu_{s_t} - \frac{1}{2} \mathbf{C}_{a_t, a_t}^{-1} \mathbf{F}_{a_t}, \\ \Sigma_{a_t} &= (-2\mathbf{C}_{a_t, a_t})^{-1} \\ &+ (-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t} \Sigma_{s_t}^T ((-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t})^T.\end{aligned}$$

We obtain,  $\begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \sim N(\mu_{a_t, s_t}, \Sigma_{a_t, s_t})$  from this:

$$\begin{aligned}\mu_{a_t, s_t} &= \begin{bmatrix} \mu_{a_t} \\ \mu_{s_t} \end{bmatrix}, \\ \Sigma_{a_t, s_t} &= \begin{bmatrix} \Sigma_{a_t} & (-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t} \Sigma_{s_t} \\ \Sigma_{s_t}^T ((-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t})^T & \Sigma_{s_t} \end{bmatrix}.\end{aligned}$$

This will be used to compute  $\mathbb{E} \left[ \sum_t \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix} \begin{bmatrix} \mathbf{a}_t \\ \mathbf{s}_t \end{bmatrix}^T \right]$  in every timestep. Similarly, it is easy to compute  $N(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$ :

$$\begin{aligned}\mu_{s_{t+1}} &= \mathbf{A} \mu_{s_t} + \mathbf{B} \mu_{a_t}, \\ \Sigma_{s_{t+1}} &= \Sigma_{error} + \\ &\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}^T \begin{bmatrix} \Sigma_{s_t} & \Sigma_{s_t}^T ((-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t})^T \\ (-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t} \Sigma_{s_t} & \Sigma_{a_t} \end{bmatrix}^T \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \\ &= \Sigma_{error} + \mathbf{A} \Sigma_{s_t}^T \mathbf{A}^T + \mathbf{B} \Sigma_{a_t}^T \mathbf{B}^T + \mathbf{B} (-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t} \Sigma_{s_t} \mathbf{A}^T \\ &+ \mathbf{A} \Sigma_{s_t}^T ((-\mathbf{C}_{a_t, a_t})^{-1} \mathbf{C}_{a_t, s_t})^T \mathbf{B}^T.\end{aligned}$$

We iteratively compute the distribution of of states, actions and state-action pairs and obtain the distribution of  $\mathbf{s}_T$ , which will be used to compute  $\mathbb{E}[(\mathbf{s}_T - \mathbf{s}_G)(\mathbf{s}_T - \mathbf{s}_G)^T]$ .