Learning Generalizable Robot Skills from Demonstrations in Cluttered Environments

M. Asif Rana, Mustafa Mukadam, S. Reza Ahmadzadeh, Sonia Chernova, and Byron Boots

***Abstract*— Learning from Demonstration (LfD) is a popular approach to endowing robots with skills without having to program them by hand. Typically, LfD relies on human demonstrations in clutter-free environments. This prevents the demonstrations from being affected by irrelevant objects, whose influence can obfuscate the true intention of the human or the constraints of the desired skill. However, it is unrealistic to assume that the robot's environment can always be restructured to remove clutter when capturing human demonstrations. To contend with this problem, we develop an *importance weighted* batch and incremental skill learning approach, building on a recent inference-based technique for skill representation and reproduction. Our approach reduces unwanted environmental influences on the learned skill, while still capturing the salient human behavior. We provide both batch and incremental versions of our approach and validate our algorithms on a 7-DOF JACO2 manipulator with *reaching* and *placing* skills.**

I. Introduction

Intelligent and cooperative robots must be capable of adapting to novel tasks in dynamic, unstructured environments. This is a challenging problem to address; it requires a robot to possess a diverse set of skills that may be difficult to hand-specify or pre-program. Learning from demonstration (LfD) has proven an effective tool in approaching such problems [1]. To acquire a desired skill, LfD approaches generally involve learning a skill model from a set of demonstrations provided by a human. The model can then be queried to reproduce the skill in novel reproduction environments with additional skill constraints. Common examples of constraints include new start/goal states, or new obstacle configurations that constrain the set of possible trajectories. LfD techniques generally differ in the manner in which the skill is represented, learned, and reproduced.

Most prior LfD approaches [2], [3], [4], [5] are based on the assumption that demonstrations can be performed in uncluttered, minimally constrained environments. The presence of clutter in the demonstration environments can introduce additional constraints on human demonstrations that are unrelated to the target skill or the underlying human intent. If unaccounted for, this can lead to suboptimal skill models. However, restructuring the world to remove clutter is often impractical, which limits the viability of such approaches.

In this work, we tackle the problem of learning skills from a set of demonstrations, which can be partially or fully influenced by the presence of obstacles (see Fig. 1).

All authors are affiliated with the Institute for Robotics and Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA. Email: {asif.rana, mmukadam3, reza.ahmadzadeh, chernova, bboots}@gatech.edu.

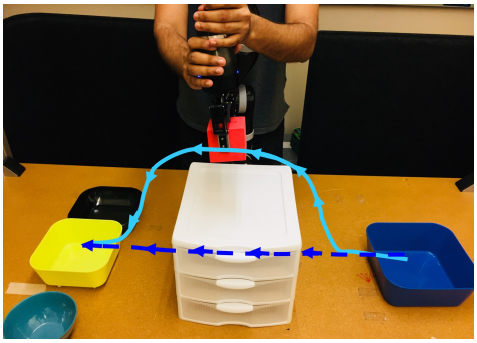


Fig. 1: A human is demonstrating a placing skill, which involves placing the (red) cube from the (blue) bowl on the right in to one of the three bowls on the left. The figure contrasts the demonstrated trajectory (light blue), which is influenced by an obstacle (drawer) in the environment, with the intended straight-line trajectory (dark blue) in the absence of the obstacle.

To contend with obstacles during training, we present *importance weighted skill learning*. Specifically, we adopt and extend the inference-based view of skill reproduction as proposed by Rana et al. [6] with Combined Learning from Demonstration And Motion Planning (CLAMP). CLAMP provides a principled approach for generalizing robot skills to novel situations, including avoiding unknown obstacles present in the reproduction environment. When reproducing a desired skill, trajectories are generated to be *optimal* with respect to the demonstrations while remaining *feasible* in the given reproduction environment.

We extend CLAMP to utilize demonstrations from cluttered environments through importance weighted skill learning (see Fig. 2), which rates the importance of demonstration trajectories while learning the skill model. We propose an importance weighting function that assigns lower importance to parts of demonstrations that are more likely to be influenced by obstacles. We present batch and incremental versions of our algorithm: batch learning is useful when the set of initial demonstrations are sufficient for learning a reasonable skill model, while incremental learning is useful in scenarios that require refinement of the skill model as new demonstrations in new environments become available.

We validate our approach on a 7-DOF JACO2 manipulator with *reaching* and *placing* skills. In all the experiments, we evaluate the approach by providing demonstrations in cluttered environments and then changing the environments for reproduction.

II. Related Work

Many existing approaches to trajectory-based LfD address the problem of avoiding obstacles in the reproduction sce-

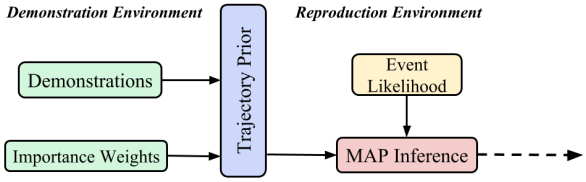


Fig. 2: An overview of our approach. In the demonstration environment, the human demonstrations and the associated importance weights are collected. The trajectory prior acts as the skill model. Conditioning this prior on the likelihood of events specified by the reproduction scenario gives the posterior.

nario. Some approaches add obstacle avoidance in the skill reproduction phase as a reactive strategy [7], [8], [9], while others carry out motion planning or trajectory optimization [10], [11], [12], [6]. In all these approaches, the skill model is learned from demonstrations that are not affected by obstacles. Any constraints or costs associated with obstacles are typically present during reproduction only. However, in an obstacle-rich environment, the demonstrations themselves are likely to be influenced by the presence of obstacles, which could have repercussions during skill reproduction.

There have been a few attempts to address the problem of learning skills from demonstrations in cluttered environments. For example, [13], [14] learn a dynamic movement primitive (DMP) as well as a coupling term for obstacle avoidance from demonstrations. These approaches suffer from two major problems. First, since DMPs follow a single demonstration, they fail to learn potentially different ways of executing the skill, thereby limiting its robustness in new scenarios. Second, due to the reactive nature of the obstacle avoidance strategy, the reproduced trajectory does not necessarily preserve the shape of the motion in the presence of obstacles. Ghalamzan et al. [15], proposed an approach based on learning a cost functional from human demonstrations. This cost functional is dependent on two components: the deviation from the mean of the demonstrations, and the distance from obstacles in the environment. Parameters of both these components are estimated from human demonstrations. A major drawback of this approach is the assumption that the mean of the demonstrations sufficiently expresses the demonstrated skill. This assumption however stands invalid for skills which can be executed in multiple ways and hence requires a more expressive skill model.

Our proposed method is based on learning an underlying stochastic dynamical system from demonstrations. Depending on the part of the state-space the robot lies in, this dynamical system is able to generate different ways of executing a learned skill. We make use of importance weighting to discount the effect of obstacles that are present when the demonstrations are provided. Specifically, the parts of demonstrations in the vicinity of obstacles are penalized to account for their deviation from the desired skill or the human intention.

III. Combined Learning from Demonstration and Motion Planning

We adopt the probabilistic inference view on learning from demonstration which has been previously employed in CLAMP [6].

A. Skill Reproduction as Probabilistic Inference

Skill reproduction using CLAMP is performed by *maximum a posteriori* (MAP) inference given a trajectory prior and event likelihoods in the reproduction environment.

Trajectory Prior: The trajectory prior or the skill model represents a distribution over robot trajectories. A trajectory is defined as a finite collection of D-dimensional robot states $\boldsymbol{x}_i \in \mathbb{R}^D$ at time t_i, $0 \leq i \leq N$. The prior is given by a joint Gaussian distribution over the robot states,

$$p(\boldsymbol{x}) \propto \exp\{-\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\mu}\|^2_{\boldsymbol{\mathcal{K}}}\}, \tag{1}$$

where,

$$\boldsymbol{x} \doteq [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T$$

$$\boldsymbol{\mu} \doteq [\boldsymbol{\mu}(t_0), \boldsymbol{\mu}(t_1), \ldots, \boldsymbol{\mu}(t_N)]^T,\ \boldsymbol{\mathcal{K}} \doteq [\boldsymbol{\mathcal{K}}(t_i, t_j)]\big|_{ij,0\leq i,j\leq N}.$$

The prior enforces *optimality* by penalizing the optimal trajectory on deviating from the mean of the prior during inference. The trajectory prior is learned from demonstrations.

Event Likelihood: The likelihood function encodes the constraints in the skill reproduction scenario. The constraints are represented as random events $\mathbf{e}$ that the optimal trajectory should satisfy thus enforcing *feasibility* during inference i.e. reproduction. These events, for example, may include obstacle avoidance, or a new start/goal state or via-point. The likelihood function is defined as a distribution in the exponential family,

$$p(\mathbf{e}|\boldsymbol{x}) \propto \exp\{-\frac{1}{2}\|\boldsymbol{h}(\boldsymbol{x};\mathbf{e})\|^2_{\boldsymbol{\Sigma}}\}, \tag{2}$$

where $\boldsymbol{h}(\boldsymbol{x};\mathbf{e})$ is a vector-valued cost function with covariance matrix $\boldsymbol{\Sigma}$. The reader is referred to [16], [6] for more details on these likelihood functions.

MAP Inference: The desired optimal and feasible trajectory that reproduces the skill is then given by,

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} \{p(\boldsymbol{x}|\mathbf{e})\} = \underset{\boldsymbol{x}}{\operatorname{argmax}} \{p(\boldsymbol{x})p(\mathbf{e}|\boldsymbol{x})\}. \tag{3}$$

B. Trajectory Prior Formulation

It is assumed that in CLAMP that robot trajectories for a desired skill are governed by an underlying linear stochastic skill dynamics,

$$\boldsymbol{x}_{i+1} = \boldsymbol{\Phi}_{i+1}\boldsymbol{x}_i + \mathbf{u}_{i+1} + \mathbf{w}_{i+1}, \quad \mathbf{w}_{i+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{i+1}), \tag{4}$$

where $\boldsymbol{\Phi}_i$ and $\boldsymbol{u}_i$ are a time-varying transition matrix and a bias term, respectively, and $\mathbf{w}_i$ is additive white noise with time-varying covariance $\mathbf{Q}_i$. The trajectory prior can be generated by taking the first and second order moments of the solution to this dynamics. This Markovian dynamics yields an exactly sparse precision matrix (inverse covariance) [17], [18] inducing structure in the trajectory prior in (1), which

enables efficient learning and inference. The problem of learning the trajectory prior is equivalent to estimating the underlying stochastic dynamics.

While learning the trajectory prior, CLAMP assumes all available demonstrations are free from external influences, and therefore captures the true human intent or skill constraints. However, in the presence of such influences, this assumption no longer holds and the learned prior is suboptimal.

IV. Importance weighted skill learning

In this section, we introduce importance weighting when learning the prior to exclude the effects of unwanted influences during demonstrations. We seek to estimate the parameters of the skill dynamics model in (4) from demonstrations. As a preliminary step, lets re-write (4) as follows,

$$\boldsymbol{x}_{i+1} = \tilde{\boldsymbol{\Phi}}_{i+1}\tilde{\boldsymbol{x}}_i + \mathbf{w}_{i+1}, \qquad \mathbf{w}_{i+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{i+1}) \tag{5}$$

where,

$$\tilde{\boldsymbol{x}}_i = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{x}_i \end{bmatrix}, \qquad \tilde{\boldsymbol{\Phi}}_{i+1} = \begin{bmatrix} \mathbf{u}_{i+1} & \boldsymbol{\Phi}_{i+1} \end{bmatrix}$$

We additionally define an importance weighting function as $w : \mathbb{R}^d \mapsto \mathbb{R}$. The importance weighting function should give higher weights to robot states that are less likely to deviate from the skill constraints or the true human intent. While this importance weighting formulation can be used in other contexts too, in this paper we define a specific form of importance weighting to account for the influence of unwanted obstacles in the demonstration environment. The exact form of this environment-dependent obstacle weighting function is presented in Section V.

A. Batch Skill Learning

Let's assume the availability of K trajectory demonstrations, with the k^{th} demonstration defined as $\boldsymbol{x}^k = [\boldsymbol{x}_0^k, \boldsymbol{x}_1^k, \ldots, \boldsymbol{x}_N^k]^T$. For each discrete time interval $(t_i, t_{i+1}]$, the inputs are collected into a matrix $\tilde{\boldsymbol{X}}_i = [\tilde{\boldsymbol{x}}_i^1, \tilde{\boldsymbol{x}}_i^2, \ldots, \tilde{\boldsymbol{x}}_i^K]$ while the corresponding targets into a matrix $\boldsymbol{X}_{i+1} = [\boldsymbol{x}_{i+1}^1, \boldsymbol{x}_{i+1}^2, \ldots, \boldsymbol{x}_{i+1}^K]$. Furthermore, the matrix $\mathbf{W}_i = \text{diag}\big(w(\boldsymbol{x}_i^1), w(\boldsymbol{x}_i^2), \ldots, w(\boldsymbol{x}_i^K)\big)$ defines a state-dependent importance weight matrix.

The batch skill learning formulation seeks to find $\tilde{\boldsymbol{\Phi}}_{i+1}$ and $\mathbf{Q}_{i+1}$, which minimize a regularized squared norm over the provided demonstrations.

$$\begin{aligned} &\tilde{\boldsymbol{\Phi}}_{i+1}^*, \mathbf{Q}_{i+1}^* \\ &\quad = \underset{\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}}{\text{argmin}} \left\{ \mathcal{L}(\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}) \right\} \\ &\quad = \underset{\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}}{\text{argmin}} \left\{ \text{tr}\big(\mathbf{Q}_{i+1}^{-1}\mathbf{E}_{i+1}\mathbf{W}_i\mathbf{E}_{i+1}^T\big) + \lambda \|\tilde{\boldsymbol{\Phi}}_{i+1}\|_F^2 \right\} \end{aligned} \tag{6}$$

where $\mathbf{E}_{i+1} = \boldsymbol{X}_{i+1} - \tilde{\boldsymbol{\Phi}}_{i+1}\tilde{\boldsymbol{X}}_i$ defines the error matrix, and λ is a regularization coefficient.

The solution to the batch skill learning problem in (6) is given by the weighted ridge regression estimate,

$$\tilde{\boldsymbol{\Phi}}_{i+1}^* = \boldsymbol{X}_{i+1}^T\mathbf{W}_i\tilde{\boldsymbol{X}}_i\big(\tilde{\boldsymbol{X}}_i\mathbf{W}_i\tilde{\boldsymbol{X}}_i^T + \lambda\mathbf{I}\big)^{-1}, \tag{7}$$

$$\mathbf{Q}_{i+1}^* = \frac{1}{z}\mathbf{E}_{i+1}^*\mathbf{W}_i{\mathbf{E}_{i+1}^*}^T, \tag{8}$$

$$z = \frac{\text{tr}(\mathbf{W}_i)^2 - \text{tr}(\mathbf{W}_i^T\mathbf{W}_i)}{\text{tr}(\mathbf{W}_i)}.$$

B. Incremental Skill Learning

The batch skill learning procedure assumes that there are enough demonstrations available to learn an optimal skill model. However, as more demonstrations are aggregated over time, possibly in different environments, it is desirable to refine the model since more data provides a better estimate of the skill. To achieve this, we propose incremental weighted skill learning.

Our incremental skill learning procedure is based on Bayesian inference. In this formulation, we maintain a joint probability distribution over the unknown skill dynamics parameters. Every time a new demonstration is collected, a posterior over the skill dynamics parameters is calculated

$$\begin{aligned} &p(\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}|\mathcal{D}^{1:k}) \\ &\quad = p(\mathcal{D}^k|\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1})p(\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}|\mathcal{D}^{1:k-1}), \end{aligned} \tag{9}$$

where $\mathcal{D}^{1:k} = \{\{\tilde{\boldsymbol{x}}_i^1, \boldsymbol{x}_{i+1}^1\}, \{\tilde{\boldsymbol{x}}_i^2, \boldsymbol{x}_{i+1}^2\}, \ldots, \{\tilde{\boldsymbol{x}}_i^k, \boldsymbol{x}_{i+1}^k\}\}$. At any stage, the mode of the posterior distribution provides an estimate of the unknown parameters.

Skill Dynamics Distribution: The joint probability distribution over the unknown parameters $\tilde{\boldsymbol{\Phi}}_{i+1}$ and $\mathbf{Q}_{i+1}$ is given by

$$p(\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}) = p(\tilde{\boldsymbol{\Phi}}_{i+1}|\mathbf{Q}_{i+1})p(\mathbf{Q}_{i+1}), \tag{10}$$

where,

$$p(\tilde{\boldsymbol{\Phi}}_{i+1}|\mathbf{Q}_{i+1}) = \mathcal{MN}(\boldsymbol{M}_{i+1}, \mathbf{Q}_{i+1}, \boldsymbol{R}_{i+1}), \tag{11}$$

$$p(\mathbf{Q}_{i+1}) = \mathcal{W}^{-1}(\boldsymbol{V}_{i+1}, \nu_{i+1}), \tag{12}$$

$\mathcal{MN}$ refers to a matrix-normal distribution with matrix-valued mean $\boldsymbol{M}_{i+1}$ and covariances $\mathbf{Q}_{i+1}$ and $\boldsymbol{R}_{i+1}$ for the rows and columns respectively. $\mathcal{W}^{-1}$ refers to an inverse-Wishart distribution with positive definite scale matrix $\boldsymbol{V}_{i+1}$ and ν_{i+1} degrees of freedom. Note that matrix-normal and inverse-Wishart distributions are generalizations of the normal and inverse-gamma distributions respectively to the multivariate case.

Demonstration Likelihood: The likelihood of observing the input-target pair from the k^{th} demonstration under the stochastic dynamics (5) is given by

$$\begin{aligned} p(\mathcal{D}^k|\tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}) &\doteq p(\boldsymbol{x}_{i+1}^k|\tilde{\boldsymbol{x}}_i^k, \tilde{\boldsymbol{\Phi}}_{i+1}, \mathbf{Q}_{i+1}) \\ &\propto \exp\left\{ -\frac{1}{2}(w_i^k\mathbf{Q}_{i+1}^{-1}\mathbf{e}_{i+1}^k{\mathbf{e}_{i+1}^k}^T) \right\}. \end{aligned} \tag{13}$$

where $\mathbf{e}_{i+1}^k = \boldsymbol{x}_{i+1}^k - \tilde{\boldsymbol{\Phi}}_{i+1}\tilde{\boldsymbol{x}}_i^k$ and $w_i^k = w(\boldsymbol{x}_i^k)$. Note that the likelihood is scaled by the weight in order to incorporate the importance weighting.

Skill Dynamics Inference: The skill dynamics parameters after assimilation of k demonstrations is given by the mode of the joint posterior distribution (*maximum a posteriori*),

$$\tilde{\mathbf{\Phi}}_{i+1}^{k}, \mathbf{Q}_{i+1}^{k} = \underset{\tilde{\mathbf{\Phi}}_{i+1}, \mathbf{Q}_{i+1}}{\operatorname{argmax}} \left\{ p(\tilde{\mathbf{\Phi}}_{i+1}, \mathbf{Q}_{i+1} | \mathcal{D}^{1:k}) \right\}. \quad (14)$$

Due to the properties of matrix-normal and inverse Wishart distributions, the mode of the joint distribution turns out to be equivalent to the product of the modes of the two conditional distributions [19],

$$\tilde{\mathbf{\Phi}}_{i+1}^{k} = \underset{\tilde{\mathbf{\Phi}}_{i+1}}{\operatorname{argmax}} \left\{ p(\tilde{\mathbf{\Phi}}_{i+1} | \mathbf{Q}_{i+1}, \mathcal{D}^{1:k}) \right\} = \boldsymbol{M}_{i+1}^{k} \quad (15)$$

$$\mathbf{Q}_{i+1}^{k} = \underset{\mathbf{Q}_{i+1}}{\operatorname{argmax}} \left\{ p(\mathbf{Q}_{i+1} | \mathcal{D}^{1:k}) \right\} = \frac{1}{\nu_{i+1}^{k} + D + 1} \boldsymbol{V}_{i+1}^{k}. \quad (16)$$

Furthermore, the parameters of the conditional distributions are governed by the following update laws,

$$\begin{aligned}
\boldsymbol{R}_{i+1}^{k} &= w_i \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^T + \boldsymbol{R}_{i+1}^{k-1} \\
\boldsymbol{M}_{i+1}^{k} &= (w_i \boldsymbol{x}_{i+1} \tilde{\boldsymbol{x}}_i^T + \boldsymbol{M}_{i+1}^{k-1} \boldsymbol{R}_{i+1}^{k-1})(\boldsymbol{R}_{i+1}^{k})^{-1} \\
\boldsymbol{V}_{i+1}^{k} &= \boldsymbol{V}_{i+1}^{k-1} + w_i (\boldsymbol{x}_{i+1} - \boldsymbol{M}_{i+1}^{k} \tilde{\boldsymbol{x}}_i)(\boldsymbol{x}_{i+1} - \boldsymbol{M}_{i+1}^{k} \tilde{\boldsymbol{x}}_i)^T \\
&\quad + (\boldsymbol{M}_{i+1}^{k} - \boldsymbol{M}_{i+1}^{k-1}) \boldsymbol{R}_{i+1}^{k-1} (\boldsymbol{M}_{i+1}^{k} - \boldsymbol{M}_{i+1}^{k-1})^T \\
\nu_{i+1}^{k} &= 1 + \nu_{i+1}^{k-1}
\end{aligned}$$

The incremental learning procedure is initialized with a prior joint distribution $p(\tilde{\mathbf{\Phi}}_{i+1}, \mathbf{Q}_{i+1} | \phi)$. The Gaussian comonent of the joint prior is selected to be the ridge regression prior, that is, $\boldsymbol{M}_{i+1}^{0} = \mathbf{0}$ and $\boldsymbol{R}_{i+1}^{0} = \frac{1}{\alpha}\boldsymbol{I}$. The inverse Wishart component is selected to be an uninformed prior, with $\boldsymbol{V}_{i+1}^{0} = \frac{1}{\beta}\boldsymbol{I}$ and $\nu_{i+1}^{0} = \frac{1}{\beta}$. Here α and β are positive scalars. In our implementation, we set $\alpha = \beta = 10^{10}$. Note that smaller values of these scalars makes the prior too strict, which restrains the skill model from fitting the data well.

V. Environment-dependent importance weighting function

In this section, we define the importance weighting function to enable skill learning from demonstrations, which may be provided in the presence of obstacles in the environment. The weighting function gives lower importance to the parts of a demonstration which are more likely to be influenced by the presence of an obstacle and therefore deviate from the intent of the human.

We hypothesize that the parts of demonstrations closer to obstacles are influenced by the obstacles and therefore fail to satisfy the skill constraints. Conversely, partial trajectories farther away from obstacles are more likely to satisfy the skill constraints and should be given more importance. For a given state $\boldsymbol{x}_i$, we define the importance weight to be equivalent to the likelihood of staying collision-free [16]. For this likelihood function, we first define a hinge loss function

$$c(\boldsymbol{x}_i) = \begin{cases} -d(\boldsymbol{x}_i) + \epsilon & d(\boldsymbol{x}_i) \leq \epsilon \\ 0 & d(\boldsymbol{x}_i) > \epsilon \end{cases},$$

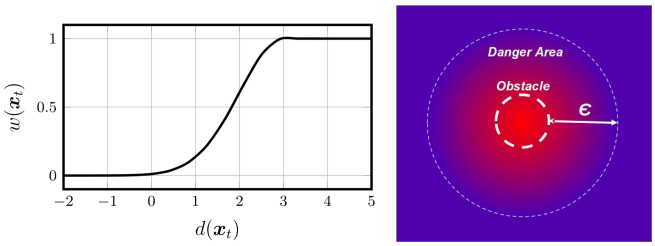


Fig. 3: An illustration of an importance weight function parameterized by $\epsilon = 3$ and $\sigma_{obs} = 1$ (left) and a signed distance field (right). The importance weight levels at 1 outside the danger area, and decays down to zero inside with the slope governed by σ_{obs}.

where $d(\cdot)$ is the signed distance from the closest obstacle in an environment and ϵ specifies the 'danger area' around the obstacle. With this hinge loss, we assume that an obstacle affects a state only when it is within the danger area around the obstacle. Outside of this danger area, the obstacle has no influence on the state. The importance weight itself is given by a function in the exponential family,

$$w(\boldsymbol{x}_i) = \exp\left\{ -\frac{c(\boldsymbol{x}_i)^2}{2\sigma_{obs}^2} \right\}, \quad (17)$$

where the parameter σ_{obs} dictates the rate of decay of the importance weight for states within the 'danger area'. The smaller the value of σ_{obs}, the faster the importance weight will decay down to zero (see Fig 3).

VI. Experiments

We evaluate the performance of our method on two different skills[1]: 1) the *reaching* skill, and 2) the *placing* skill. For both skills, a human provides multiple demonstrations via kinesthetic teaching on a 7-DOF JACO2 manipulator. The end-effector positions are recorded and the corresponding instantaneous velocities are estimated by fitting a cubic spline to each demonstration and taking its time derivative. Furthermore, the demonstrations are also time-aligned using dynamic time warping (DTW). To setup the trajectory prior in (1), we define the robot states $\boldsymbol{x}_i$ as the vector concatenation of instantaneous robot positions and velocities.

For the *reaching* skill, the goal is to reach an object from different locations. Hence, all the demonstrations share the same goal state while the initial state varies. In the absence of any obstacles in the path, a demonstration follows a nearly straight-line path to the goal. In the presence of obstacles in the path, the demonstrations deviate from this desired path in order to avoid collision with the obstacles. Fig. 4 shows the demonstration environment and the corresponding demonstrations.

In order to learn the trajectory prior for this skill, we use importance weighted skill learning, as described in Section IV-A. The demonstrations reaching the target from the uncluttered part of the environment represent the true human intent. Therefore, we expect our trajectory prior to be biased towards these demonstrations. Fig. 5 shows the trajectory distributions (i.e. time-evolving state distributions)

[1]Accompanying video: https://youtu.be/03r8Tblhq7k

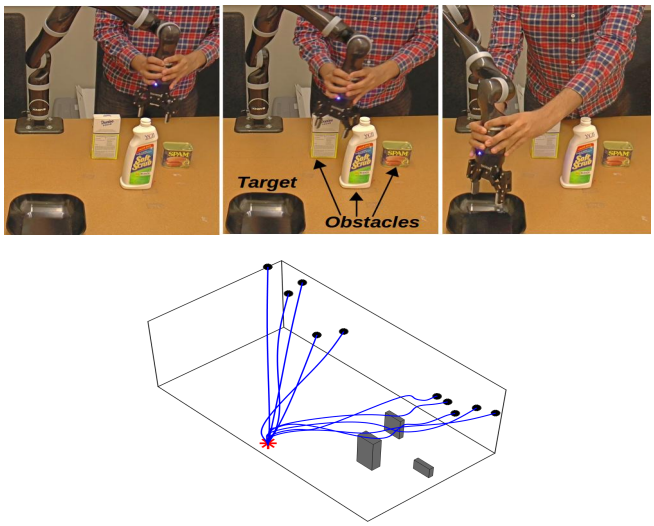
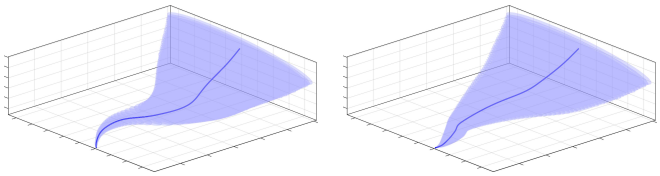


Fig. 4: Human demonstrations for the *reaching* skill. All demonstrations reach the bowl from different initial positions in the presence of three obstacles in the environment. *Top*: Snapshots of a demonstrations avoiding the obstacles. *Bottom*: A 3-D plot showing all the demonstrations and the obstacles.



(a) without importance weighting (b) with importance weighting

Fig. 5: Trajectory prior visualization for the *reaching* skill. The blue line is the mean of the prior, and the blue shaded region shows one standard deviation around the mean.

encoded in the trajectory priors learned with and without importance weighting. The trajectory distributions are generated by rolling out the stochastic skill dynamics in (5) with an initial state distribution given by a Gaussian over the initial demonstration states. The mean of the trajectory distribution generated with importance weighting deviates less from the intended straight-line path, exhibiting the true underlying skill, as compared to the distribution without importance weighting. To enable this, we empirically selected the parameters of the importance weight function in (17), such that the parts of state-space likely to be under obstacle influence can be successfully downplayed while learning the prior. A value of $\epsilon = 0.3m$ and $\sigma_{obs} = 0.01m$ provided sufficient bounding region around the obstacles in most cases.

Fig. 6 shows multiple instances of reproduction for the *reaching* skill. The skill is reproduced with (3) by conditioning the learned trajectory prior on the likelihood of starting from a desired initial state and the likelihood of staying clear of arbitrarily placed obstacles. We show the trajectories generated from two different initial states in three different environments. When the obstacles are placed at the same location as the demonstration phase or displaced, the reproduced trajectories from the prior without importance weighting take the longer path to the target around the

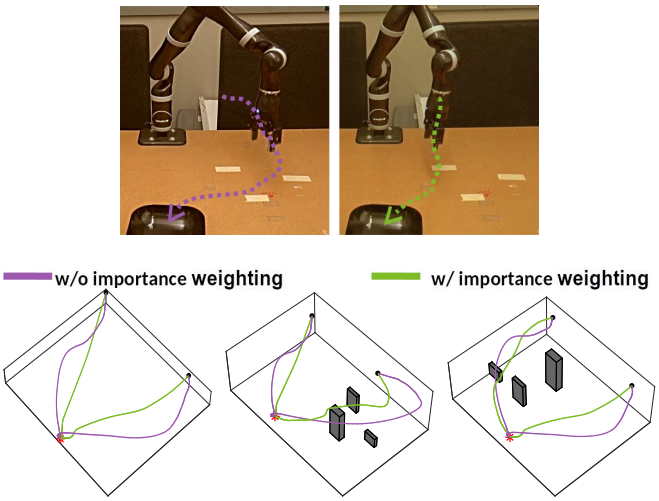


Fig. 6: Trajectories generated by conditioning the priors on two initial positions in three different environments. *Top-left*: Environment without obstacles. *Top-center*: Environment with obstacles at the same locations as demonstrations. *Top-right*: Environment with obstacles displaced. *Bottom*: Trajectory executions on a real robot in the obstacle-free environment.

obstacles. This is because the demonstrations on average took a longer path while avoiding obstacles and the prior shown in Fig. 5(a) forces the reproduced trajectories to exhibit a similar behavior. For the same reasons, the deviant non-smooth trajectories are also observed when no obstacles are present in the vicinity of the robot in the reproduction environment.

The *placing* skill involves placing an object at different locations on a table. All the demonstrations start from the same location since the object's initial location is fixed. The end state of the demonstration varies with the target placement location. Initially there is an obstacle present in the desired path, hence all the demonstrations go above the obstacle causing them to be influenced. Fig. 7 (left) plots the human demonstrations provided in this scenario. Since only the influenced demonstrations are available at this stage, the trajectory prior learned from these demonstrations also encodes the influence of obstacles which is undesirable. However, as the environment changes and more demonstrations are available in a cleaner environment, as shown in Fig. 7 (right), the prior is updated using the incremental weighted learning procedure described in Section IV-B.

Fig. 8 shows the evolution of the prior as demonstrations are assimilated. The prior initially enforces highly constrained motion causing the trajectories to avoid the obstacle even when it is not present. As more demonstrations are made available in an obstacle-free environment, the high importance weight relative to the influenced demonstrations enables adaptation to the desired underlying motion after just three updates. On the other hand, when the importance weighting is not considered in the incremental learning procedure, the trajectory prior still exhibits the obstacle influence even after all the demonstrations are incorporated. This is shown in Fig. 9. The utility of the incremental learning procedure is high in such scenarios. It is undesirable

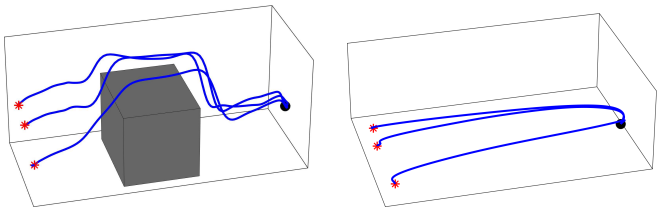


Fig. 7: Human demonstrations for the *placing* skill in two different environments. *Left*: Environment with a large obstacle influencing the demonstrations. *Right*: Obstacle-free environment.

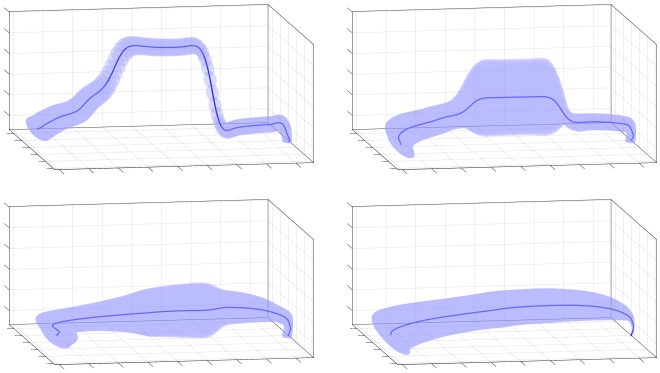


Fig. 8: Trajectory priors for the *placing* skill with importance weighting. *Top-left*: Learned from first 3 demonstrations recorded in the presence of obstacle. *Top-right*: Prior after assimilating fourth demonstration in clean environment. *Bottom-left*: Prior after assimilating fifth demonstration. *Bottom-right*: Final prior after all the incremental updates.

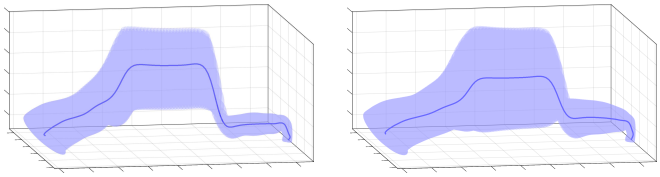


Fig. 9: Trajectory priors for the *placing* skill without importance weighting. *Left*: Learned after assimilating first 4 demonstrations. *Right*: Final prior after all the incremental updates.

to keep all the demonstrations and re-learn the prior on arrival of each new demonstration, since this can be both time-consuming as well as memory-intensive.

VII. Conclusion

We have presented *importance weighted skill learning*, which is a novel technique for learning skills from demonstrations in cluttered environments and generalizing them to new scenarios. Our importance weighting function associates lower weights with parts of demonstrations that are likely to collide with obstacles. We conjecture that demonstrations which are in close proximity to obstacles are more susceptible to not satisfying the constraints of the skill being learned. Hence, those demonstrations should be given lesser importance during the skill learning stage. Our learning approach is also capable of incrementally updating and refining the skill model to incorporate new demonstrations without the need to relearn the model from scratch. Since our learning method is based on extracting the underlying stochastic skill dynamics, it does not share the same disadvantages as approaches that assume a mean trajectory to encode the skill. Furthermore, our reproduction method is capable of generalizing the skill efficiently across various scenarios as demonstrated in the experiments.

Acknowledgements

This research is supported in part by NSF NRI 1637758, NSF CAREER 1750483, NSF IIS 1637562, and ONR N00014-16-1-2844.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[3] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[4] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[5] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in neural information processing systems*, 2013, pp. 2616–2624.

[6] M. A. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots, "Towards robust skill generalization: Unifying learning from demonstration and motion planning," in *Conference on Robot Learning*, 2017, pp. 109–118.

[7] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *2009 IEEE International Conference on Robotics and Automation(ICRA)*. IEEE, 2009, pp. 763–768.

[8] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *8th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2008, pp. 91–98.

[9] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, 2012.

[10] G. Ye and R. Alterovitz, "Demonstration-guided motion planning," in *International Symposium on Robotics Research (ISRR)*, vol. 5, 2011.

[11] T. Osa, A. M. G. Esfahani, R. Stolkin, R. Lioutikov, J. Peters, and G. Neumann, "Guiding trajectory optimization by demonstrated distributions," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 819–826, 2017.

[12] D. Koert, G. Maeda, R. Lioutikov, G. Neumann, and J. Peters, "Demonstration based trajectory optimization for generalizable robot motions," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 515–522.

[13] A. Rai, F. Meier, A. Ijspeert, and S. Schaal, "Learning coupling terms for obstacle avoidance," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 512–518.

[14] A. Gams, M. Denisa, and A. Ude, "Learning of parametric coupling terms for robot-environment interaction," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 304–309.

[15] A. Ghalamzan, C. Paxton, G. D. Hager, and L. Bascetta, "An incremental approach to learning generalizable robot tasks from human demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5616–5621.

[16] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots, "Continuous-time Gaussian process motion planning via probabilistic inference," *arXiv preprint arXiv:1707.07383*, 2017.

[17] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 9–15.

[18] T. Barfoot, C. H. Tong, and S. Sarkka, "Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression," *Proceedings of Robotics: Science and Systems, Berkeley, USA*, 2014.

[19] T. Minka, "Bayesian linear regression," Citeseer, Tech. Rep., 2000.