Active Learning of Bayesian Probabilistic Movement Primitives

Thibaut Kulak, Hakan Girgin, Jean-Marc Odobez and Sylvain Calinon

Abstract—**Learning from Demonstration permits non-expert users to easily and intuitively reprogram robots. Among approaches embracing this paradigm, probabilistic movement primitives (ProMPs) are a well-established and widely used method to learn trajectory distributions. However, providing or requesting useful demonstrations is not easy, as quantifying what constitutes a good demonstration in terms of generalization capabilities is not trivial. In this paper, we propose an active learning method for contextual ProMPs for addressing this problem. More specifically, we learn the trajectory distributions using a Bayesian Gaussian mixture model (BGMM) and then leverage the notion of epistemic uncertainties to iteratively choose new context query points for demonstrations. We show that this approach reduces the required number of human demonstrations. We demonstrate the effectiveness of the approach on a pouring task, both in simulation and on a real 7-DoF Franka Emika robot.**

Index Terms—**Imitation Learning, Learning from Demonstration, Incremental Learning**

I. INTRODUCTION

LEARNING from demonstration (LfD) offers an intuitive framework for non-expert users to easily (re)program robots. One well-established LfD approach is called probabilistic movement primitives (ProMPs) [1], which permits movement representation and generation. ProMPs have been successfully used for learning different robotic tasks from demonstrations, including rhythmic tasks [2], striking tasks [3], or human-robot collaboration tasks [4]. One of the main capabilities of ProMPs lies in the task generalization, which is usually achieved by conditioning the trajectory distribution to some desired keypoints. It is also desirable and possible to generalize with respect to a context or external variable, which is known before executing the task (such as the mass of an object or the volume of a liquid to pour), by learning the joint distribution of the context variable and the trajectory [5, 6]. Task generalization is crucial for robotic applications. This requires a set of demonstrations to provide various executions of the task, whose acquisition is often costly. Thus, we want to collect these demonstrations in an efficient manner. Often, non-expert users struggle to identify what demonstration will be the most informative to the robot [7]. One way to alleviate this limitation is to provide the user with some feedback, such as a visual illustration of what the robot has currently learned [8]. Yet, such an approach requires the appropriate design of a feedback mechanism, which might not be trivial in a high-dimensional task, and still requires the user to choose the demonstration eventually. In contrast, we propose to automatically determine what constitutes a good demonstration.

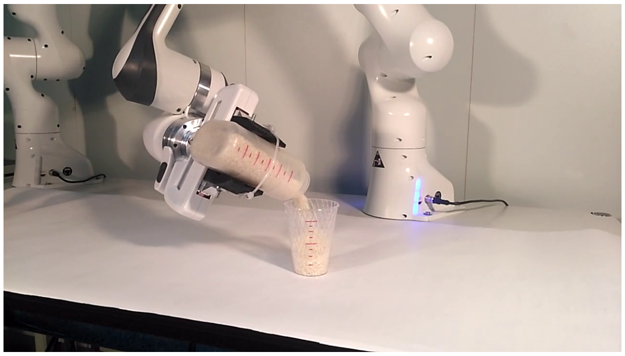


Fig. 1: Overview of the pouring task with a 7-axis robot.

Active learning is a promising approach as it allows the robot to actively request a demonstration to improve its comprehension of the task. This alleviates the human burden of choosing which demonstration to provide, and is expected to reduce the number of demonstrations required for effective generalization. The main component of an active learning framework is a metric allowing to select the demonstration that is expected to yield the greatest improvement. Traditionally, this metric is based on uncertainties [9]. When building statistical models, two different kinds of uncertainties arise, namely *aleatoric uncertainties* and *epistemic uncertainties*. Aleatoric uncertainties represent the variations in the demonstrations, i.e., different possible ways to perform the task. This is the uncertainty that is captured by ProMPs when fitting a Gaussian or a Gaussian mixture model (GMM) to the demonstrations. Such uncertainty is then typically used to define when the robot must be stiff and where it can be compliant. In contrast, epistemic uncertainties represent the uncertainties due to the lack of data. In other words, aleatoric uncertainties cannot be reduced by adding more data, while epistemic uncertainties can be. For this reason, the quantification of epistemic uncertainties is crucial for active learning frameworks.

In this paper, we propose an active learning approach for ProMPs with the aim of improving the generalization capabilities by relying on fewer demonstrations. To do so, we use Bayesian inference [10] to quantify both aleatoric and epistemic uncertainties in ProMPs. Specifically, we propose

Manuscript received: 10, 15, 2020; Revised: 01, 07, 2021; Accepted: 01, 26, 2021.

This paper was recommended for publication by Dana Kulic upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the ROSALIS project (Swiss National Science Foundation) and by the COLLABORATE project (https://collaborate-project.eu/), funded by the EU within H2020-DT-FOF-02-2018 under grant agreement 820767.

The authors are with the Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland (e-mail: firstname.lastname@idiap.ch.)

Digital Object Identifier (DOI): see top of this page.

to learn the ProMP with a Bayesian Gaussian mixture model (BGMM). In Sec. III, we introduce Bayesian ProMPs. Then, in Sec. IV, we propose an active learning method based on the epistemic uncertainties captured by the BGMM. We demonstrate the applicability of our approach in Sec. V on three different pouring task experiments. The first two experiments are performed in simulation to allow quantitative comparisons and for reproducibility purposes. The last experiment shows the applicability of the approach on a real 7-DoF robot pouring task.

The contributions of this paper are threefold: *(i)* we propose a principled methodology for deriving epistemic uncertainties in ProMPs; *(ii)* we propose to use a closed-form lower bound of the differential entropy of the epistemic uncertainty as an information gain metric for an active learning of ProMPs; *(iii)* we show the applicability of the approach on a robotic pouring task.

II. RELATED WORK

As the data acquisition process is usually costly in robotics, active learning has emerged as a viable solution. It has been shown that active learning permits a faster exploration of the action space [11], which is particularly true in the context of developmental robotics, where active learning is often referred to as curiosity-driven learning [12, 13]. In the context of learning from demonstrations, active imitation learning [14] is a topic gaining interest. It has indeed been successfully used in a variety of robotic tasks, such as autonomous navigation [15, 16]. In [17], the authors leverage the uncertainties on a discrete hypothesis space to request meaningful demonstrations to a human teacher. Several approaches have also been proposed in the context where the learner does not request full demonstrations, but only the action to take at a given state [14, 18]. In [19], the authors propose to use active learning with BGMMs to learn control policies from demonstrations, and show the effectiveness of the approach on a reaching task with obstacles. One important limitation of this work is that the uncertainties are computed for an action given the current state. Hence, it is not applicable to robotic tasks where one needs to reason about the uncertainty over the whole task (e.g., over the whole trajectory), which is often the case in robotics (for instance for object grasping, assembly or navigation tasks). Also, the method requires the possibility to start and show a demonstration from any given state, which is not always possible (for instance, starting a demonstration in the middle of a dynamic throwing task or a pouring task is not feasible).

Closer to our work is [20], in which Gaussian process regression (GPR) is used to learn a trajectory given a context. It is applied to a reaching task where the context is the desired end-effector position. Although Gaussian processes are very efficient for capturing epistemic uncertainties, they do not capture aleatoric uncertainties (variations of the task). It is therefore not applicable to tasks where one wants to use the aleatoric uncertainties for compliant control. As there is no guarantee of convergence of the retrieved trajectory to the desired final location, they combine the trajectory predicted by GPR with a dynamic movement primitive (DMP) approach that attracts the robot to the desired goal. Thus, their approach might not be applicable for tasks where the context is not the desired end-effector position.

In [21] the authors propose an active learning method for learning ProMPs. The distribution is learned in the ProMP weight space using a GMM. They then use the marginal distribution over the internal context space (trajectory keypoint) to request demonstrations for contexts that are the furthest from any Gaussian (as Mahalanobis distance). Their approach is evaluated for a reaching task where different grasps are possible, with attempts to generalize over different poses of the object. This approach has several limitations. First, they choose the next context to query based only on some distance in the context space. While in their application this can make sense since the contexts (keypoints) are closely correlated with the trajectory distributions, this is not relevant for a more general external context. Indeed, representing the context space well is not so useful, as our ProMPs are used to generate trajectory distributions for a given context. Rather, what matters is whether a given context influences the trajectory distribution. In this regard, their method would aim to represent a context variable with no influence on trajectories equally well as other more meaningful context variables. In contrast, our method focuses on the conditional distribution of the weights given the context, hence learning dependencies and correlations between the context variables and the movement. A second limitation is that the use of a GMM does not take into account epistemic uncertainties but only aleatoric ones, while work in active learning [9] has shown that metrics based on aleatoric uncertainties are less effective than those based on epistemic uncertainties. Lastly, their approach uses a heuristics to add Gaussians during learning using a threshold. Indeed, the Mahalanobis distance does not depend on the weights attributed to the different Gaussians, which might bias the learning towards unlikely portions of the context space. In contrast, we use Bayesian inference to infer the number of Gaussians using a Dirichlet prior on the mixing coefficients.

III. BAYESIAN PROMPS

In this section, we present the BGMM framework for learning contextual ProMPs.

A. Contextual ProMP

A ProMP is a probability distribution over trajectories built from a series of N demonstrations (trajectories) of length T and of D dimensions. A demonstration $\boldsymbol{\tau} \in \mathbb{R}^{(T\times D)}$ is approximated by a sum of M basis functions, which are often chosen as radial basis functions (RBF)

$$\boldsymbol{\tau}_i = \boldsymbol{\Phi}\boldsymbol{w}_i + \boldsymbol{\epsilon}, \qquad \text{with} \qquad \boldsymbol{\Phi} = \boldsymbol{\Phi}^{\text{1d}} \otimes \mathbb{I}_D, \tag{1}$$

where $\otimes$ represents the Kronecker product, $\boldsymbol{\epsilon}$ is zero-mean i.i.d. Gaussian noise, $\boldsymbol{w}_i$ of size $MD\times 1$ is the weight associated to the i^{th} demonstration, $\boldsymbol{\Phi}^{\text{1d}}_{T\times M}$ is the basis function matrix with $\boldsymbol{\Phi}^{\text{1d}}_{t,m} = \Phi_m(t)$ corresponding to the m^{th} basis function indexed at time t, and $\mathbb{I}_D$ is an identity matrix. The weight vectors associated to each demonstration are computed with least squares as

$$\boldsymbol{w}_i = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\tau}_i. \tag{2}$$

A probability distribution $p(\boldsymbol{w})$ can then be learned from the demonstrations $\{\boldsymbol{w}_i\}_{i=1}^N$, usually with a multivariate Gaussian or a GMM.

We focus on tasks where adaptation with respect to an external context variable is required. Such context variable can be any environmental property such as an object mass, an object position, or the amount of liquid in a pitcher for a pouring task. Note that the method is general and would be applicable to internal context variables as well (e.g., trajectory keypoints). A common way [2, 5] to take into account context variables is to learn the joint distribution of contexts and weights $p(\boldsymbol{c}, \boldsymbol{w})$, where $\boldsymbol{c}$ is the context variable of size D^c. For notation convenience, we introduce $\tilde{\boldsymbol{w}}_i = [\boldsymbol{c}_i^\top, \boldsymbol{w}_i^\top]^\top$, hence $p(\boldsymbol{c}, \boldsymbol{w}) = p(\tilde{\boldsymbol{w}})$.

B. Problem formulation

The goal of the task lies in how to modulate the movement $\boldsymbol{w}$ based on different contexts $\boldsymbol{c}$. We denote the context space $\mathcal{C}$ as the space of all possible contexts we would like our robot to be able to generalize to. Formally, this means that there exists an unknown ground truth target distribution $p^{\text{GT}}(\boldsymbol{c}, \boldsymbol{w})$ that can be used to generate robot movements $p^{\text{GT}}(\boldsymbol{w}|\boldsymbol{c})$ adapted for context $\boldsymbol{c}$. We aim to learn this unknown joint distribution by active imitation learning.

C. Bayesian Gaussian Mixture Model (BGMM)

In this section, we present the learning of the joint distribution of contexts and weights with a BGMM using variational inference. The joint distribution is defined with a mixture of K multivariate normal distributions (MVNs) with means $\boldsymbol{\mu}{=}\{\boldsymbol{\mu}_k\}_{k=1}^K$, precision matrices $\boldsymbol{\Lambda}{=}\{\boldsymbol{\Lambda}_k\}_{k=1}^K$ and mixing coefficients $\boldsymbol{\pi}{=}\{\pi_k\}_{k=1}^K$ as

$$p(\tilde{\boldsymbol{w}}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \sum_{k=1}^K \pi_k \mathcal{N}(\tilde{\boldsymbol{w}}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}).$$

A Normal-Wishart prior is used for means and precision matrices, and a Dirichlet prior is put on the mixing coefficients:

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{k=1}^K \mathcal{N}(\boldsymbol{\mu}_k|(\beta_0\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{S}_k, \nu_k), \tag{3}$$

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\alpha_0). \tag{4}$$

The means, the precision matrices and the mixing coefficients maximizing the posterior distribution are estimated using closed-form update equations similar to those of the Expectation-Maximization algorithm for the maximum likelihood solution, see Section 10.2.1 in [10] for further details. Also, they are available as parts of standard machine learning libraries (e.g., *scikit-learn* for Python).

Given N demonstrations $\tilde{\boldsymbol{W}} = \{\tilde{\boldsymbol{w}}\}_{i=1}^N$, the predictive density of a new pair of context and weight $\hat{\tilde{\boldsymbol{w}}} = [\hat{\boldsymbol{c}}^\top, \hat{\boldsymbol{w}}^\top]^\top$ is equivalent to a mixture of multivariate t-distributions with mean $\hat{\boldsymbol{m}}_k$, covariance matrix $\hat{\boldsymbol{L}}_k$, mixing coefficients $\hat{\pi}_k$ and degrees of freedom $\hat{\nu}_k$

$$p(\hat{\tilde{\boldsymbol{w}}}|\tilde{\boldsymbol{W}}) = \sum_{k=1}^K \hat{\pi}_k \ \text{t}(\hat{\tilde{\boldsymbol{w}}}|\hat{\boldsymbol{m}}_k, \hat{\boldsymbol{L}}_k, \hat{\nu}_k), \qquad \text{where}$$

$$\begin{aligned} \hat{\pi}_k &= \frac{\alpha_k}{\sum_{k=1}^K \alpha_k}, \\ \hat{\nu}_k &= \nu_k{+}1{-}D{-}D^c, \\ \hat{\boldsymbol{L}}_k &= \frac{(\nu_k{+}1{-}D{-}D^c)\beta_k}{1{+}\beta_k}\boldsymbol{S}_k, \\ \hat{\boldsymbol{m}}_k &= \boldsymbol{m}_k, \end{aligned} \tag{5}$$

where α_k, β_k and $\boldsymbol{m}_k$ are derived from statistics of the data. We do not include the full equations here, but the reader can refer to Equations 10.41–10.63 of [10] for more details.

We can then condition on the context to get the conditional posterior predictive distribution of the weights for a given context variable as in [10] (Section 10.2.3)

$$p(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}}) = \sum_{k=1}^K \hat{\pi}_k^{w|c} \ \text{t}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{m}}_k^{w|c}, \hat{\boldsymbol{L}}_k^{w|c}, \hat{\nu}_k^{w|c}), \tag{6}$$

$$\text{with} \quad \hat{\pi}_k^{w|c} = \frac{\hat{\pi}_k \ \text{t}(\hat{\boldsymbol{c}}|\hat{\boldsymbol{m}}_k^c, \hat{\boldsymbol{L}}_k^c, \nu_k^c)}{\sum_{j=1}^K \hat{\pi}_j \ \text{t}(\hat{\boldsymbol{c}}|\hat{\boldsymbol{m}}_j^c, \hat{\boldsymbol{L}}_j^c, \nu_j^c)}, \tag{7}$$

$$\hat{\nu}_k^{w|c} = \hat{\nu}_k{+}D^c, \tag{8}$$

$$\hat{\boldsymbol{m}}_k^{w|c} = \hat{\boldsymbol{m}}_k^w{+}\hat{\boldsymbol{L}}_k^{wc}\hat{\boldsymbol{L}}_k^{cc^{-1}}(\hat{\boldsymbol{c}}{-}\hat{\boldsymbol{m}}_k^c), \tag{9}$$

$$\hat{\boldsymbol{L}}_k^{w|c} = \frac{\hat{\nu}_k{+}(\hat{\boldsymbol{c}}{-}\hat{\boldsymbol{m}}_k^c)^\top\hat{\boldsymbol{L}}_k^{cc^{-1}}(\hat{\boldsymbol{c}}{-}\hat{\boldsymbol{m}}_k^c)}{\hat{\nu}_k^{w|c}} \cdot (\hat{\boldsymbol{L}}_k^{ww}{-}\hat{\boldsymbol{L}}_k^{wc}\hat{\boldsymbol{L}}_k^{cc^{-1}}\hat{\boldsymbol{L}}_k^{wc^\top}), \tag{10}$$

where we have decomposed $\hat{\boldsymbol{L}}_k = \begin{bmatrix} \hat{\boldsymbol{L}}_k^{cc} & \hat{\boldsymbol{L}}_k^{wc^\top} \\ \hat{\boldsymbol{L}}_k^{wc} & \hat{\boldsymbol{L}}_k^{ww} \end{bmatrix}$.

We have shown how contextual ProMPs can be learned with Bayesian GMMs. We will now propose an active learning strategy leveraging the uncertainties learned by the Bayesian model.

IV. ACTIVE LEARNING OF PROMPS

In this section, we propose an active learning strategy for Bayesian ProMPs. First, we show how aleatoric and epistemic uncertainties can be separated when conditioning. Then, we propose a closed-form information gain metric based on the entropy of the conditional distribution. Finally, the full active learning process is summarized.

A. Uncertainty decomposition

The conditional posterior predictive distribution of the Bayesian ProMP encodes two types of uncertainties: the aleatoric uncertainty (possible variations of the task, the one learned with standard ProMPs) and the epistemic uncertainty (representing the lack of knowledge). Indeed, from Eq. (10), we can see that the covariance matrix of the conditional posterior predictive distribution can be decomposed into two parts (see also [19])

$$\hat{\boldsymbol{L}}_k^{w|c} = \hat{\boldsymbol{L}}_k^{\text{al}}{+}\hat{\boldsymbol{L}}_k^{\text{ep}}, \qquad \text{where} \tag{11}$$

$$\hat{\boldsymbol{L}}_k^{\mathrm{al}} = \frac{\hat{\nu}_k}{\hat{\nu}_k^{w|c}}(\hat{\boldsymbol{L}}_k^{ww} - \hat{\boldsymbol{L}}_k^{wc}\hat{\boldsymbol{L}}_k^{cc^{-1}}\hat{\boldsymbol{L}}_k^{wc^\top}), \tag{12}$$

$$\hat{\boldsymbol{L}}_k^{\mathrm{ep}} = \frac{(\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^c)^\top \hat{\boldsymbol{L}}_k^{cc^{-1}}(\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^c)}{\hat{\nu}_k^{w|c}}(\hat{\boldsymbol{L}}_k^{ww} - \hat{\boldsymbol{L}}_k^{wc}\hat{\boldsymbol{L}}_k^{cc^{-1}}\hat{\boldsymbol{L}}_k^{wc^\top}). \tag{13}$$

Notice that the first part does not depend on the context $\hat{\boldsymbol{c}}$, while the second part grows quadratically with it. This was observed in [10] (Section 3.3.2) for Bayesian linear regression. In that sense, we argue that the first part can be attributed to the aleatoric uncertainty, and the second to the epistemic uncertainty. Indeed, the first part cannot be reduced when adding more data as it models the variability in the demonstrations due to the fact that for the same given context $\hat{\boldsymbol{c}}$ different movements can be executed to achieve the task. On the other hand, the second term can be reduced when having more data. Actually, in the limit where the amount of data and the number of Gaussians would grow to infinity, the context space would be perfectly represented and the term $(\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^c)^\top \hat{\boldsymbol{L}}_k^{cc^{-1}}(\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^c)$ would tend to zero. In practice, the above decomposition is particularly useful in the context of ProMPs, because we can have access to the aleatoric uncertainty to design compliant behaviors, or to the epistemic uncertainty for quantifying the lack of knowledge of the model.

B. Uncertainty measurement

The most general and common uncertainty measure is the Shannon entropy [22]. Initially proposed for discrete random variables, the Shannon entropy has been extended to continuous probability distributions, in which case it is called continuous (or differential) entropy. We propose to quantify the uncertainty of our conditional ProMP by calculating the (continuous) entropy of its epistemic part.

The entropy of a mixture of multivariate t-distributions cannot be obtained analytically. To avoid computationally expensive Monte Carlo sampling methods, we propose to approximate the distribution with a GMM, for which there is a closed-form lower bound of the entropy. The epistemic part of the conditional ProMP distribution can be approximated by a mixture of K Gaussians using moment matching:

$$\tilde{\pi}_k(\boldsymbol{c}) = \hat{\pi}_k^{w|c},\ \tilde{\boldsymbol{\mu}}_k(\boldsymbol{c}) = \hat{\boldsymbol{m}}_k^{w|c},\ \tilde{\boldsymbol{\Sigma}}_k(\boldsymbol{c}) = \frac{\hat{\nu}_k^{w|c}}{\hat{\nu}_k^{w|c} - 2}\hat{\boldsymbol{L}}_k^{\mathrm{ep}}(\boldsymbol{c}). \tag{14}$$

We propose to use the closed-form lower bound introduced in [23], which has been shown to be tight. It is expressed as (for clarity purposes we omit the fact that all GMM parameters depend on $\boldsymbol{c}$)

$$H_{lower}(p^{ep}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}})) = \frac{1}{2}\Big(K \log 2\pi + K + \sum_{i=1}^{K} \tilde{\pi}_i \log|\tilde{\boldsymbol{\Sigma}}_i|\Big) - \sum_{i=1}^{K} \tilde{\pi}_i \log \sum_{j=1}^{K} \tilde{\pi}_j e^{-C_\alpha(p_i, p_j)}, \tag{15}$$

where $C_\alpha(p_i, p_j)$ is the Chernoff α-divergence distance function between the i^{th} and j^{th} Gaussians for $\alpha \in [0, 1]$:

$$C_\alpha(p_i, p_j) = \frac{(1-\alpha)\alpha}{2} \cdot (\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j)^\top \Big((1-\alpha)\tilde{\boldsymbol{\Sigma}}_i + \alpha\tilde{\boldsymbol{\Sigma}}_j\Big)^{-1}(\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j) + \frac{1}{2}\log\left(\frac{|(1-\alpha)\tilde{\boldsymbol{\Sigma}}_i + \alpha\tilde{\boldsymbol{\Sigma}}_j|}{|\tilde{\boldsymbol{\Sigma}}_i|^{1-\alpha}|\tilde{\boldsymbol{\Sigma}}_j|^{\alpha}}\right). \tag{16}$$

In practice we choose $\alpha = 1/2$, in which case the Chernoff divergence is the Bhattacharyya distance.

The full active learning process is summarized in Algorithm 1. Finding the context which maximizes the epistemic entropy can be done either using a grid search if the context space is of low dimension, or using a Bayesian optimization algorithm.

Algorithm 1: Choosing the demonstration context.

```
Data: demonstrations $\tilde{\boldsymbol{W}} = \{\boldsymbol{c}_i, \boldsymbol{w}_i\}_{i=1}^N$, context
      search space $\mathcal{C}$
Result: context $\boldsymbol{c}^*$ at which to request a
        demonstration

Learn joint distribution of $p(\boldsymbol{c}, \boldsymbol{w}) = p(\tilde{\boldsymbol{w}})$ with
 BGMM;
Calculate $p(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}})$ using Equations (6) to (10);
Isolate the epistemic uncertainty
 $p^{ep}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}})$ with Equations (11) and (13);
Approximate the entropy of
 $p^{ep}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}})$ with Equations (14) to (16);

Find $\boldsymbol{c}^* = \arg\max_{\hat{\boldsymbol{c}} \in \mathcal{C}} H_{lower}(p^{ep}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}}))$
```

V. EXPERIMENTS

In this section, we evaluate our active learning method in four different ways related to the pouring task. The first three favor quantitative results and reproducibility by using a simulated environment and a given database of demonstrations to choose from. In the last experiment, we consider the pouring task on a real 7 DoF Franka Emika robot.

In all experiments, we use $N = 20$ evenly spread Gaussian radial basis functions (RBFs) for ProMP. The width of the RBFs are set as $h = (\frac{T-1}{N})^2$. The hyperparameters of the BGMM are the default hyperparameters of the *scikit-learn* library. We choose a diagonal covariance matrix prior, with a standard deviation of 0.1 for the context variables and 1 for the ProMP weights. We use a maximum number of 5 Gaussians, or strictly less than the number of demonstrations if there are less than 6 demonstrations.

Throughout the experiments, we compare our method to three baselines. The first one (**Random**) is a random strategy using the same BGMM representation as our method. The second one (**GP**) is an adaptation of [20] for external context

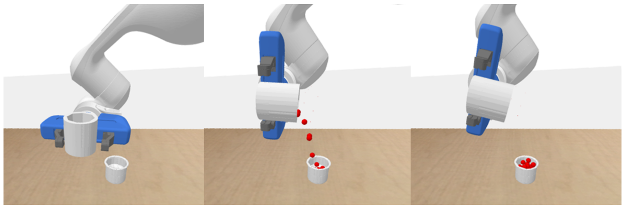


Fig. 2: Overview of the simulated pouring environment.

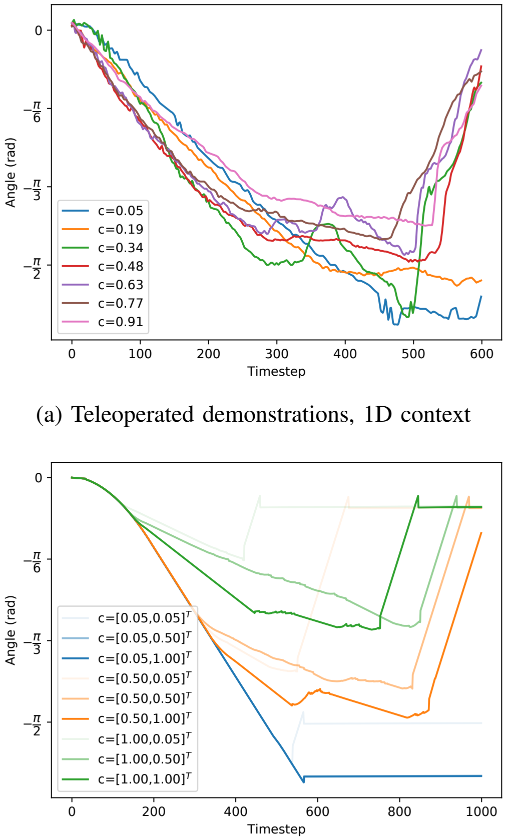
variables: we learn the conditional model of the trajectories given the context with a Gaussian process (GP)[1] using a squared exponential kernel (hyperparameters optimization gave a length scale of 1 and output variance of 0.1^2). The active learning approach for the GP baseline selects the context for which the conditional distribution of the trajectories given the context has the most variance. The third baseline (**Conkey19**) is an adaptation of [21] (introduced in Sec. II) for external context variables: we learn the joint distribution of contexts and ProMP weights with a GMM and use the Mahalanobis distance in the context space as an active learning measure. We use the same covariance prior as with our approach, and we use $\beta = 3$ for the hyperparameter governing how many outliers are discarded when adding a new datapoint to the Gaussian mixture, see Eq. (7) of [21] for more details[2].

A. Simulated pouring

We use here a simulated pouring environment implementing the Franka Emika robot in the PyBullet simulator [24]. The goal of this task is to pour liquid (simulated as rigid spherical particles because PyBullet does not support fluids simulation) from a pitcher into a mug. An overview of the simulated setup is shown in Fig. 2. In the first two simulated environments, we avoid learning the affordances of the object and control directly the orientation of the edge of the pitcher, from where the liquid is poured. This permits us to make the task with a reference trajectory of just one variable: the angle of the pitcher. In the third simulated environment, we go beyond the one-dimensional control angle case, and show the robustness of our approach for more complex movements encoded in a 6-dimensional control variable.

1) 1D context: In this first experiment, we consider a one-dimensional context variable, which represents the amount of liquid in the pitcher. As the mug volume is lower than the pitcher volume, one difficulty of the task is to stop pouring when the mug is full. We consider context variables varying from 0.05 to 1, representing how full the pitcher is (from 5% to 100%). In this experiment, the goal is to fill the mug completely (without overflowing).

In order to have demonstrations exhibiting realistic variations, we provide real human demonstrations using teleoperation. As the reference trajectory contains only a one-dimensional angle, teleoperation is made simply using a camera by detecting the angle of a colored object held by the human demonstrator. We build a dataset of 100 demonstrations for contexts evenly spread between 0.05 and 1. Namely, we choose $\mathcal{C} = \{0.05 + \frac{1-0.05}{99}k\}_{k=0}^{99}$ and provide one teleoperated demonstration for each context in $\mathcal{C}$. This permits reproducibility of the results and a fair comparison of the methods as they have access to the same demonstrations for given contexts. Demonstrations are aligned using linear interpolation. A subset of aligned demonstrations is shown in Fig. 3a. We can effectively see that, the more the pitcher is filled, the less it has to be tilted to pour into the mug. We start the active learning process with 2 initial demonstrations, for contexts randomly chosen in the context space $\mathcal{C}$. We make the experiment 20 times with different initial demonstrations. We show in Fig. 4 how it compares to a random strategy which randomly chooses the next context. In Fig. 4a, we plot the mean epistemic entropy (averaged on the context space $\mathcal{C}$) in function of the number of requested demonstrations. We can see that our strategy outperforms the random strategy in terms of reduction of the epistemic uncertainties. The diminution of the epistemic uncertainty is particularly big during the first 5 demonstrations requested with our method. In Fig. 4b, we propose an objective metric for comparing quantitatively the two methods. We introduce the task cost, which is simply a ℓ_2 norm between the final volume in the mug and the desired final volume (approximated with the number of balls in the mug. The desired number of balls is 50, which corresponds to the mug being almost completely filled. Filling it too



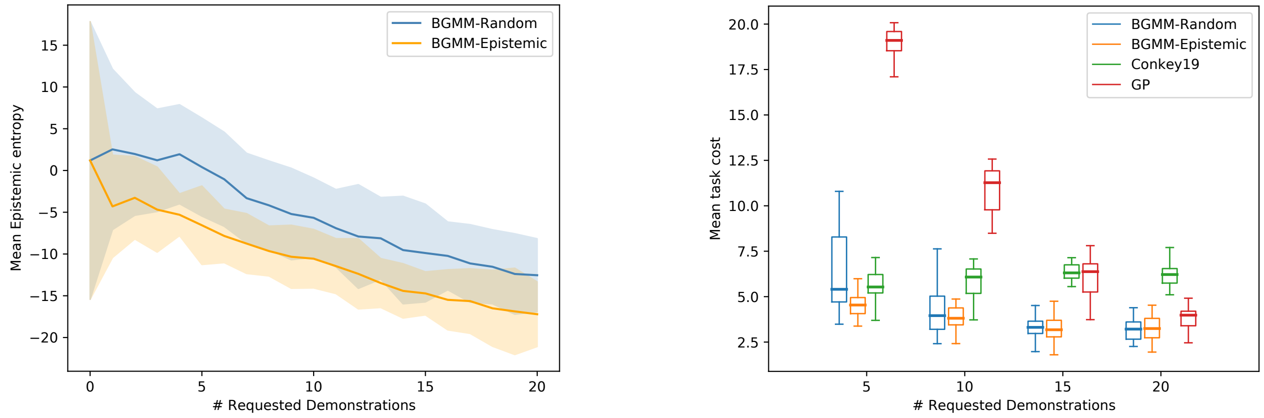
(a) Teleoperated demonstrations, 1D context

(b) Generated demonstrations, 2D context

Fig. 3: Subset of demonstrations for different contexts.

[1] Alternatively, we could also learn a GP from contexts to ProMP weights, but in practice it gave the same results as learning directly from contexts to trajectories. For this reason, we do not include it in this paper.

[2] Authors advised to choose β between 2 and 3, we chose 3 because it gave the best results.



(a) Reduction of epistemic uncertainty w.r.t number of demonstrations

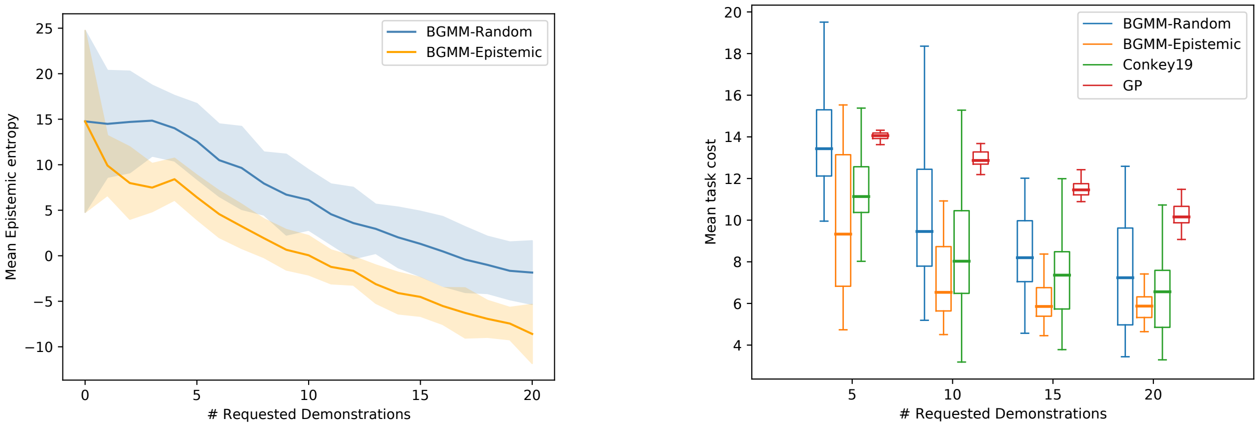
(b) Reduction of task cost w.r.t number of demonstrations

Fig. 4: Quantitative results for simulated 1D context pouring.

much is possible and increases the task cost as well). We observe in Fig. 4b that our method significantly outperforms the random strategy in the beginning of the learning process (5 demonstrations), while afterwards the results are similar. This suggests that our active learning strategy improves learning with few demonstrations. As the context is low-dimensional (1 dimension), this is not surprising that for more than 10 demonstrations, active learning does not yield any improvement over a random strategy which has also explored the context space well. It is also interesting to note that our method has less variance across experiments than the random strategy. Also, our movement representation with a BGMM gives much better results than the GP approach as it achieves a significantly lower task cost at all stages of the learning process. We can see that our method also outperforms Conkey19, whose performance stagnates during the learning process. We believe this is due to the heuristics that are proposed to add Gaussians to the mixture, which had only been tested in the 2D case in the original paper, and that would probably need to be adjusted.

2) 2D context: In this experiment we propose to add another context variable: the desired final volume in the mug. This context variable also ranges from 0.05 to 1, representing how full the mug is (from 5% to 100%). We then have $c = [c_{\text{pitcher}}, c_{\text{mug}}]^T$. For this task, we manually implement a controller performing the task, which is used as the human demonstrator (note that the demonstrations may not be perfect, e.g., when there is not enough liquid in the pitcher initially to fill the mug to its desired level). A sample of generated demonstrations can be found in Fig. 3b. We can see that, for a given desired volume in the mug, the smaller the initial volume of the pitcher is, the more the pitcher needs to be tilted. And, for a given initial volume of the pitcher, the more the mug needs to be filled, the more the object has to be tilted. Note that we do not bring the pitcher back to its horizontal position when it is fully emptied. As in the previous experiment, for reproducibility reasons, we precompute a database of generated demonstrations. A grid of width 20 is used to represent the context space for which demonstrations are generated, yielding 400 demonstrations. Namely, $\mathcal{C} = \{(0.05+\frac{1-0.05}{99}i, 0.05+\frac{1-0.05}{99}j)\}_{i,j=0}^{19}$. We also perform 20 experiments where each experiment starts with 2 randomly sampled demonstrations from the database. Results are shown in Fig. 5. We can see in Fig. 5a that our strategy outperforms the random strategy in terms of reduction of the epistemic uncertainties. More importantly, we see in Fig. 5b that the active learning strategy can learn the task using fewer demonstrations than a random strategy. Namely, the model improved with 5 demonstrations obtained using our method achieves lower task cost than if the same model was improved with 10 demonstrations using the random strategy. Similarly, 10 actively gathered demonstrations contribute better to the task cost than 20 randomly gathered ones. This shows that the entropy of the epistemic uncertainties of a BGMM is a good metric for actively learning ProMPs. We also observe that our BGMM approach significantly outperforms the GP baseline. In particular, we see that the GP approach is on par with the BGMM-Random approach after 5 requested demonstrations, but then performs worse than the two approaches based on BGMMs. This motivates the use of our Bayesian representation based on ProMPs for learning robot movements, instead of a Gaussian Process approach. Note also that our approach has the additional advantage of quantifying the aleatoric uncertainty as well, which can typically be exploited in ProMPs for designing compliant controllers. Also, we observe that in this experiment the Conkey19 approach performs similarly to our approach, though slightly worse. As explained in the previous subsection, we believe this is because this approach was developed for a 2-dimensional context case.

3) 3D context: In this experiment, we want to test the robustness of our method with respect to higher-dimensional context and control variables. Hence, we add a third context variable related to the position where the pitcher was grasped by the robot. Namely, the robot always starts from the same position but the pitcher can have been grasped at different heights between the base and the top. This makes the movement more complex as one rotation angle is not sufficient anymore to characterize it, and there are correlations between the robot translations and rotations. We use a 6-dimensional control variable consisting of position and orientation (Euler angles) of the robot end-effector. A controller is implemented



(a) Reduction of epistemic uncertainty w.r.t number of demonstrations

(b) Reduction of task cost w.r.t number of demonstrations

Fig. 5: Quantitative results for simulated 2D context pouring.

to execute the task, and is used as the human demonstrator. For this experiment, due to the higher dimensionality of the context space, we do not precompute a database of demonstrations as in previous experiments but generate online the demonstrations requested by the algorithm, and use a Bayesian optimization algorithm (the tree-structured Parzen estimator approach [25] implemented in the *hyperopt* Python package [26]) to calculate the context yielding the highest epistemic entropy.

We can see in Fig. 6a that the reduction of the epistemic uncertainties is bigger with our active learning metric than with the random baseline, similarly to what we observed in the past two experiments, and that this epistemic reduction correlates with a better task cost error (see Fig. 6b), confirming that the epistemic uncertainties seem to be a good active learning metric. Finally, our method outperforms the two alternative baselines from the literature by a very large margin in this more complicated experiment.

B. Real robot pouring task

In this experiment we demonstrate the viability of our approach on a pouring task with a real 7-axis Franka Emika Panda robot. An overview of the physical setup can be seen in Fig. 1. The context space is 2-dimensional as in the previous simulated experiment, with context variables ranging from 10% to 100%. In this experiment, we also show the robustness of our approach to several degrees of freedom as we choose the demonstrations to be 3-dimensional (position in the vertical plane containing the pitcher and the glass, and orientation of the pitcher). We give 2 initial demonstrations to the robot in random contexts, and the robot iteratively requests 20 additional demonstrations. The first 3 iterations of the active learning process are shown in Fig. 7. We can see that the robot starts by requesting demonstrations at the corners of the state space, which is normal because this is where it is the most uncertain. Note that we could use an information-density method to make the requests close to the demonstrations (e.g., by adding a similarity objective). We verified qualitatively that the learned movement representation permits to pour successfully for different contexts, which can be seen on the supplementary video (we tested it on 9 different contexts, taken from a 3×3 grid in the context space).

VI. Conclusion

In this paper, we proposed to use Bayesian Gaussian mixture models to learn ProMPs. We introduced a closed-form entropy measure leveraging the epistemic uncertainties captured by the Bayesian model. We demonstrated the usefulness of the approach both in simulation and on the real robot, showing that it reduces the number of demonstrations required to learn a movement representation that has good generalization capabilities. In future work, we plan to find metrics for determining how many demonstrations are required, and extend this method for combining LfD and reinforcement learning of ProMPs in a unified active learning framework.

References

[1] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 1–9.

[2] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[3] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters, "Using probabilistic movement primitives for striking movements," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, 2016, pp. 502–508.

[4] G. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks," *Autonomous Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[5] M. Ewerton, G. Maeda, G. Kollegger, J. Wiemeyer, and J. Peters, "Incremental imitation learning of context-dependent motor skills," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, 2016, pp. 351–358.

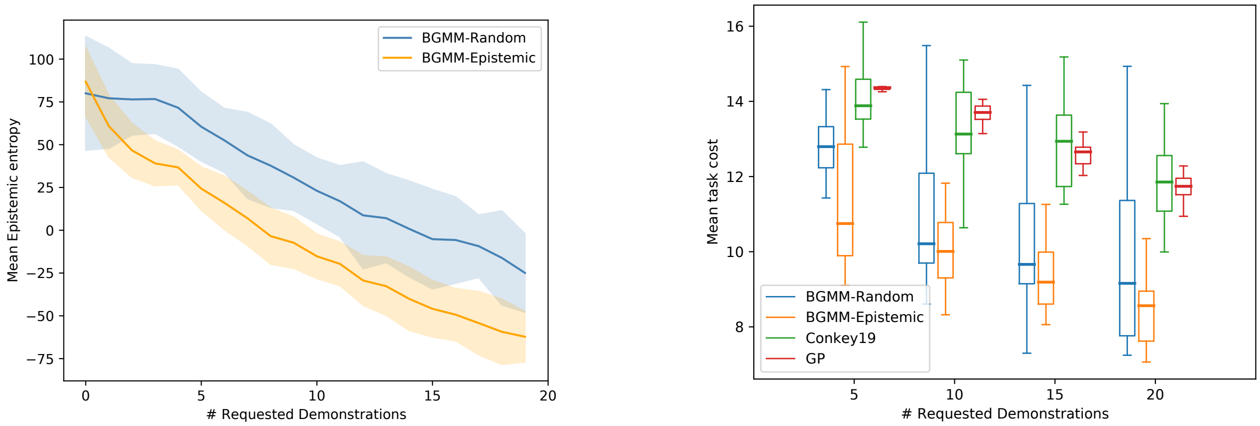
[6] A. Paraschos, E. Rueckert, J. Peters, and G. Neumann, "Probabilistic movement primitives under unknown system dynamics," *Advanced Robotics*, vol. 32, no. 6, pp. 297–310, 2018.

[7] A. Sena, Y. Zhao, and M. Howard, "Teaching human teachers to teach robot learners." in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2018, pp. 5675–5681.

[8] A. Sena and M. Howard, "Quantifying teaching behavior in robot learning from demonstration," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 54–72, 2020.

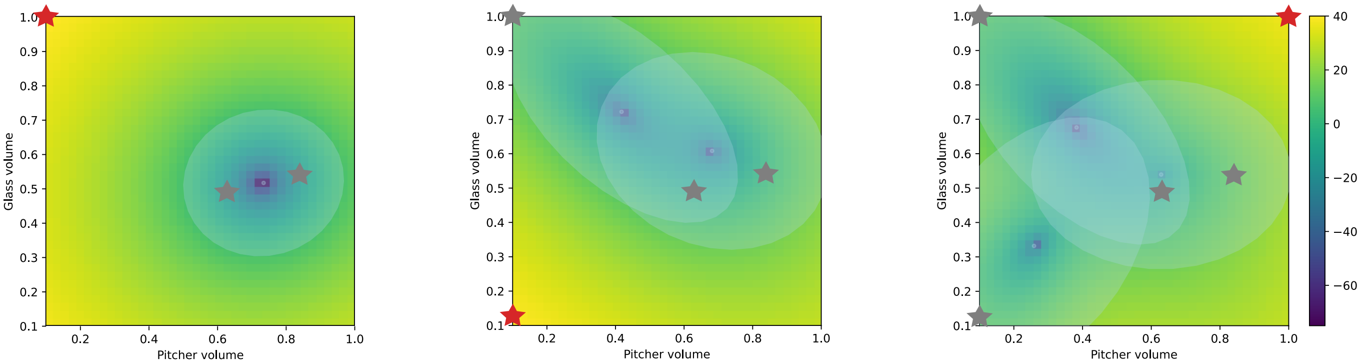
[9] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.



(a) Reduction of epistemic uncertainty w.r.t number of demonstrations (b) Reduction of task cost w.r.t number of demonstrations

Fig. 6: Quantitative results for simulated 3D context pouring.



(a) First iteration (b) Second iteration (c) Third iteration

Fig. 7: Visualization of the context space during the first 3 iterations of the active learning process. The heatmap represents the entropy of the epistemic uncertainty, yellow indicating high uncertainty. Demonstrations are shown as grey stars. The context chosen for the next demonstration is shown as a red star. Transparent ellipses show the marginal distribution of the ProMP in the context space.

[11] M. Salganicoff, L. Ungar, and R. Bajcsy, "Active learning for vision-based robot grasping," *Machine Learning*, vol. 23, no. 2-3, pp. 251–278, 1996.

[12] S. Ivaldi, N. Lyubova, A. Droniou, D. Gerardeaux-Viret, D. Filliat, V. Padois, O. Sigaud, P. Oudeyer, *et al.*, "Learning to recognize objects through curiosity-driven manipulation with the icub humanoid robot," in *Proc. IEEE Intl Conf. on Development and Learning and Epigenetic Robotics (ICDL)*, 2013, pp. 1–8.

[13] J. Schmidhuber, "Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts," *Connection Science*, vol. 18, no. 2, pp. 173–187, 2006.

[14] A. Shon, D. Verma, and R. Rao, "Active imitation learning," in *Proc. AAAI Conference on Artificial Intelligence*, 2007, pp. 1–7.

[15] D. Silver, J. Bagnell, and A. Stentz, "Active learning from demonstration for robust autonomous navigation," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2012, pp. 200–207.

[16] O. Kroemer, R. Detry, J. Piater, and J. Peters, "Combining active learning and reactive control for robot grasping," *Robotics and Autonomous systems*, vol. 58, no. 9, pp. 1105–1116, 2010.

[17] C. Chao, M. Cakmak, and A. Thomaz, "Transparent active learning for robots," in *Proc. ACM/IEEE Intl Conf. on Human-Robot Interaction (HRI)*, 2010, pp. 317–324.

[18] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, pp. 1–25, 2009.

[19] H. Girgin, E. Pignat, N. Jaquier, and S. Calinon, "Active improvement of control policies with Bayesian Gaussian mixture model," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 5395–5401.

[20] G. Maeda, M. Ewerton, T. Osa, B. Busch, and J. Peters, "Active incremental learning of robot movement primitives," in *Conference on Robot Learning (CoRL)*, vol. 78, 2017, pp. 37–46.

[21] A. Conkey and T. Hermans, "Active learning of probabilistic movement primitives," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, 2019, pp. 1–8.

[22] C. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[23] A. Kolchinsky and B. Tracey, "Estimating mixture entropy with pairwise distances," *Entropy*, vol. 19, no. 7, p. 361, 2017.

[24] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2020.

[25] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

[26] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in science conference*, vol. 13, 2013, p. 20.