Active Improvement of Control Policies with Bayesian Gaussian Mixture Model

Hakan Girgin, Emmanuel Pignat, Noémie Jaquier and Sylvain Calinon

***Abstract*— Learning from demonstration (LfD) is an intuitive framework allowing non-expert users to easily (re-)program robots. However, the quality and quantity of demonstrations have a great influence on the generalization performances of LfD approaches. In this paper, we introduce a novel active learning framework in order to improve the generalization capabilities of control policies. The proposed approach is based on the epistemic uncertainties of Bayesian Gaussian mixture models (BGMMs). We determine the new query point location by optimizing a closed-form information-density cost based on the quadratic Rényi entropy. Furthermore, to better represent uncertain regions and to avoid local optima problem, we propose to approximate the active learning cost with a Gaussian mixture model (GMM). We demonstrate our active learning framework in the context of a reaching task in a cluttered environment with an illustrative toy example and a real experiment with a Panda robot.**

I. INTRODUCTION

Learning from demonstration (LfD) offers an intuitive framework to overcome the difficulty of programming robots by teaching them movements using an adaptive representation. One of the main LfD approaches is called behavior cloning or policy imitation, and consists in inferring the parameters of a movement model via supervised learning [1]–[7] from a demonstration dataset. In LfD, the demonstrations are often acquired by kinesthetic teaching or by teleoperation. One of the main advantages of these techniques is that they allow non-expert users to easily (re-)program the robots. However, it may not be straightforward to determine the number of demonstrations necessary for the robot to learn a specific skill, as well as the locations in which they should be provided. Moreover, acquiring the demonstrations can be costly especially in industrial environments. Therefore, we aim at collecting these demonstrations in an informative way.

Active learning is a promising approach to address the aforementioned issues. An active learning framework develops and tests new hypotheses in an interactive learning process. In robotics, the robot is first provided with initial demonstrations from which an initial model of the task can be built. Then, at each stage of the active learning framework, the robot is expected to request a new demonstration in order to improve the model. This contrasts with passive learning systems that attempt to explain the model only according to available training data. Ideally, the robot should request the new demonstration around a query point that will maximize the information gain. Specifically, the information gain is related to the areas where the model uncertainties are the highest.

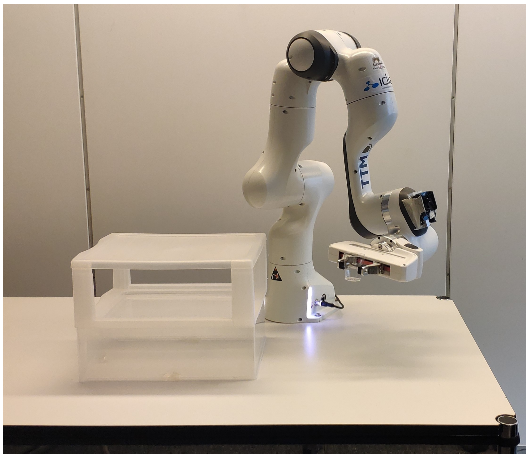


Fig. 1: Experimental setup with Franka Emika Panda robot. The task is to put the cup inside a box covered from top and bottom, starting from anywhere in the space. The robot has to maintain a specific end-effector orientation to perform the task without pouring the cup. The main challenge is not to collide with the box and the other obstacles in the environment.

In robotics applications, two different kinds of uncertainties arise, namely *(i) aleatoric uncertainties* and *(ii) epistemic uncertainties*. The aleatoric uncertainties represent the variations in the demonstrations and are typically used to adapt the behavior of the robot, e.g. its compliance at different phases of the task. In contrast, the epistemic uncertainties are related to the lack of knowledge (i.e. data) in the demonstrations and is typically used for informative exploration. Active learning is thus based on the epistemic uncertainties in the model. A natural way to take these uncertainties into account is through Bayesian inference [8].

In this work, we propose an active learning approach with the aim of improving the generalization capabilities of control policies in a behavior cloning setup, also called policy imitation. Our approach is based on the framework presented in [9] which models a joint distribution $p(\boldsymbol{x}_t, \boldsymbol{u}_t)$ in action-state abstraction with Bayesian Gaussian mixture models (BGMMs). The conditional (predictive) distribution of the policy $p(\boldsymbol{u}_t|\boldsymbol{x}_t)$ is then found by conditioning on the current state $\boldsymbol{x}_t$. In [9], the authors use a product of

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

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

experts (PoE) framework to exploit the uncertainties inherent to Bayesian models in order to fuse several control policies (see Section III for a brief background). The proposed active learning approach is based on the epistemic uncertainties in BGMM model. A method to decompose the covariance matrix of the posterior BGMM distribution into aleatoric and epistemic parts is first presented in Section IV-A. The quadratic Rényi entropy is then used to compute the related uncertainties of Gaussian mixture models (GMMs) in closed-form (see Section IV-B). As explained in Section IV-C, the next query point of our active learning framework is obtained by maximizing an information-density cost based on the quadratic Rényi entropies. In particular, we propose to approximate this cost with a GMM to represent highly uncertain region distribution. This notably avoids local optima problem during uncertainty maximization. Finally, we demonstrate the efficiency of our approach on a reaching task in a cluttered environment in a 2D simulated example and with a real experiment on a Panda robot (see Section V). The experiment setup is presented by Fig. 1.

The contributions of this paper are threefold: *(i)* we provide an uncertainty decomposition in BGMM control policies to be used for exploitation and exploration in behavior cloning approaches; *(ii)* we introduce an information weighted closed-form cost to describe uncertain regions of the state space; and *(iii)* we propose an active learning framework which can be used with partial demonstrations, with closed-form monitoring of the uncertainty reduction.

II. RELATED WORK

A collection of recent work focuses on improving and fine-tuning learned movement representations using reinforcement learning (RL) [10, 11] and iterative learning control (ILC) [12]. As these methods iteratively minimize a reward function, LfD can be used to determine the initial point of the optimization in order to favor a safe exploration. In contrast, information-theoretic explorations in behavior cloning methods have been exploited only in few works to enhance the quality and the generalization abilities of the learned movement models [13]–[15].

One of the simplest and widely used active learning methods is uncertainty sampling. Using an uncertainty measure, the robot is expected to request a query point in the most uncertain region of the input space. If the model can only encode aleatoric uncertainties, one can train several probabilistic representations with different local convergence properties. The disagreement between each individual model and their average model is then maximized using KL divergence as explained in [16]. Other techniques consist in reducing the variance of error in a regression problem. In general, this is intractable. Simplifications occur by using Fisher Information and Cramér-Rao inequality as in [17]. All the aforementioned methods are myopic as they only care about the information content of single data instances. This can result in models selecting outliers or exploring far away in the context space where no generalization is required. Information-density methods overcome this problem by choosing instances that have high information content and are still representative of the underlying distribution. This is achieved by using a weighted product of uncertainty measure (entropy, ensemble, etc) and similarity measure (Euclidean distance, correlation coefficients, etc.) [16].

In [13], the authors use Gaussian Process Regression (GPR) in a reaching task to map object positions to the weights encoding the trajectories via Dynamic Movement Primitives (DMP). They demonstrate an active learning framework based on the GPR epistemic uncertainties for reaching to a predefined set of object positions to improve their DMP model. They work with time-dependent trajectory policies without control information. They measure the epistemic uncertainty of a whole trajectory given a context, while aleatoric uncertainties (variations) are not considered. Our work differs from [13] in two ways. First, as we consider state-dependent policies including both aleatoric and epistemic uncertainties. Second, their approach in [13] exploits uncertainty sampling, which would diverge if the uncertainty is defined over a continuous variable instead of a discrete set of variables. To overcome this problem, we use information-density methods.

III. BACKGROUND

In this section, we present the BGMM framework exploited to learn control policies presented in [9]. As state-dependent control policies learned with BGMM can create unstable behaviors, the BGMM policy is fused with another stable control policy within the PoE framework.

A. Bayesian Gaussian Mixture Model

In this section, the Bayesian analysis of a Gaussian Mixture Model (GMM) is treated following [8]. Let $\boldsymbol{x} = \left[\boldsymbol{x}^{i\top}\, \boldsymbol{x}^{o\top}\right]^\top \in \mathbb{R}^D$ be the joint observation of the input and the output with dimension $D = D_i{+}D_o$. The joint distribution is defined with a mixture of K multivariate normal distributions (MVNs) with means $\boldsymbol{\mu}{=}\{\boldsymbol{\mu}_k\}$, precision matrices $\boldsymbol{\Lambda}{=}\{\boldsymbol{\Lambda}_k\}$ and mixing coefficients $\boldsymbol{\pi}{=}\{\pi_k\}$ as

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

We define a latent variable $\boldsymbol{z}$, each component of which is a binary variable $z_k \in \{0,1\}$ such that $\sum_{k=1}^{K} z_k = 1$. We can associate the mixing coefficients to the latent variables with $p(z_k{=}1) = \pi_k$ so that $p(\boldsymbol{z}|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_k}$. We then obtain $p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_k}$. The conditional distributions $p(\boldsymbol{Z}|\boldsymbol{\pi})$, $p(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{\mu},\boldsymbol{\Lambda})$, the conjugate prior distributions $p(\boldsymbol{\mu},\boldsymbol{\Lambda})$ and $p(\boldsymbol{\pi})$ of the joint observation dataset $\boldsymbol{X}{=}\{\boldsymbol{x}_n\}$ and the latent variable dataset $\boldsymbol{Z}{=}\{\boldsymbol{z}_n\}$ are summarized in Table I.

As explained in [8], closed-form update equations for Expectation-Maximization (EM) algorithm is derived by using a factorized variational distribution. Note that EM update equations are usually implemented in machine learning libraries such as *scikit-learn* for Python.

For robotic applications, we determine the predictive density of a new observation point $\hat{\boldsymbol{x}} = \left[\hat{\boldsymbol{x}}^i\, \hat{\boldsymbol{x}}^o\right]^\top$ equivalent

TABLE I: conditionals and priors where $\mathcal{W}(\cdot)$ and $\mathrm{Dir}(\cdot)$ correspond to Wishart and Dirichlet distributions

Conditional of $\boldsymbol{X}$ $p(\boldsymbol{X}\vert\boldsymbol{Z},\boldsymbol{\mu},\boldsymbol{\Lambda})$	$\prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{x}_n\vert\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k^{-1})^{z_{nk}}$
Conditional of $\boldsymbol{Z}$ $p(\boldsymbol{Z}\vert\boldsymbol{\pi})$	$\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}$
Prior on $\boldsymbol{\mu},\boldsymbol{\Lambda}$ $p(\boldsymbol{\mu},\boldsymbol{\Lambda})$	$\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k\vert\boldsymbol{m}_0,(\beta_0\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k\vert\boldsymbol{W}_0,\nu_0)$
Prior on $\boldsymbol{\pi}$ $p(\boldsymbol{\pi})$	$\mathrm{Dir}(\boldsymbol{\pi}\vert\alpha_0)$

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 degree of freedoms $\hat{\nu}_k$ as [8]

$$p(\hat{\boldsymbol{x}}|\boldsymbol{X}) = \sum_{k=1}^{K}\pi_k \mathrm{t}(\hat{\boldsymbol{x}}|\boldsymbol{m}_k, \boldsymbol{L}_k, \nu_k), \tag{1}$$

where

$$\pi_k = \frac{\alpha_k}{\sum_{k=1}^{K}\alpha_k}, \tag{2}$$

$$\nu_k = \nu_k + 1 - D, \tag{3}$$

$$\boldsymbol{L}_k = \frac{(\nu_k + 1 - D)\beta_k}{1+\beta_k}\boldsymbol{W}_k, \tag{4}$$

$$\boldsymbol{m}_k = \bar{\boldsymbol{m}}_k. \tag{5}$$

with the update equations on α_k,β_k ν_k, $\boldsymbol{W}_k$ and $\bar{\boldsymbol{m}}_k$ are given ?? in [8]. We can then define the distribution of the output conditioned on the input as

$$p(\hat{\boldsymbol{x}}^o|\hat{\boldsymbol{x}}^i, \boldsymbol{X}) = \sum_{k=1}^{K}\pi_k^{o|i}\mathrm{t}(\hat{\boldsymbol{x}}^i|\boldsymbol{m}_k^{o|i}, \boldsymbol{L}_k^{o|i}, \nu_k^{o|i}), \tag{6}$$

where

$$\pi_k^{o|i} = \frac{\pi_k \mathrm{t}(\hat{\boldsymbol{x}}^i|\boldsymbol{m}_k^i, \boldsymbol{L}_k^i, \nu_k^i)}{\sum_{j=1}^{K}\pi_j \mathrm{t}(\hat{\boldsymbol{x}}^i|\boldsymbol{m}_j^i, \boldsymbol{L}_j^i, \nu_j^i)}, \tag{7}$$

$$\nu_k^{o|i} = \nu_k + D^i, \tag{8}$$

$$\hat{m}_k^{o|i} = \boldsymbol{m}_k^o + \boldsymbol{L}_k^{oi}{\boldsymbol{L}_k^{ii}}^{-1}(\hat{\boldsymbol{x}}^i - \boldsymbol{m}_k^i), \tag{9}$$

$$\boldsymbol{L}_s = \boldsymbol{L}_k^{oo} - \boldsymbol{L}_k^{oi}{\boldsymbol{L}_k^{ii}}^{-1}{\boldsymbol{L}_k^{oi}}^{\top}, \tag{10}$$

$$L_k^{o|i} = \frac{\nu_k + (\hat{\boldsymbol{x}}^i - \boldsymbol{m}_k^i)^{\top}{\boldsymbol{L}_k^{ii}}^{-1}(\hat{\boldsymbol{x}}^i - \boldsymbol{m}_k^i)}{\nu_k^{o|i}}\boldsymbol{L}_s. \tag{11}$$

In this work, we consider the input $\hat{\boldsymbol{x}}^i$ and the output $\hat{\boldsymbol{x}}^o$ equivalent to the state $\boldsymbol{x}$ and the control command $\boldsymbol{u}$, respectively. Note that the stability of this controller is determined by the positive-definiteness of the term $\boldsymbol{L}_k^{oi}{\boldsymbol{L}_k^{ii}}^{-1}$. To guarantee the controller stability, the PoE framework is introduced in the next section.

B. Product of Experts

Robot movements learned with state-action abstractions result in probabilistic controllers with no guarantee of stability, unless explicitly constrained to be stable as in [6]. To overcome this problem, we fuse the probabilistic unstable controller with another probabilistic stable controller which acts as an attractor towards the demonstration area when the uncertainty in the unstable controller is high. We refer to this fusion of controllers as a *product of experts* (PoE), where each expert represents a stochastic controller with different uncertainty properties. Note that many types of controllers with different uncertainties can be fused to work in parallel. For more details, we refer the reader to [9].

In this work, the stabilizing controller is defined as a probabilistic linear quadratic tracker policy, which can be expressed as a MVN. It can be viewed as a controller which attracts the system to the demonstrated regions when the BGMM controller is very uncertain. When the BGMM control policy is a GMM, the fusion or PoE is defined as the product of a GMM and a MVN, which results in another GMM policy. As an illustrative example, consider a 2D reaching task in a cluttered environment. Fig. 2a displays the initial demonstrations starting from different initial positions (cross) to reach goal position ($\boldsymbol{G}$). We choose 5 different random test initial positions and reproduce the trajectories by sampling from a BGMM model and a PoE model. The resulting trajectories are shown in Fig. 2b and 2c, respectively. Even though the trajectories are more stable in 2c (notice that some of the trajectories in 2b diverge), the task cannot be accomplished without colliding with the obstacles. In this case, supplementary demonstrations are necessary, and active learning permits to collect them in an informed way.

IV. ACTIVE LEARNING WITH BAYESIAN GAUSSIAN MIXTURE MODEL

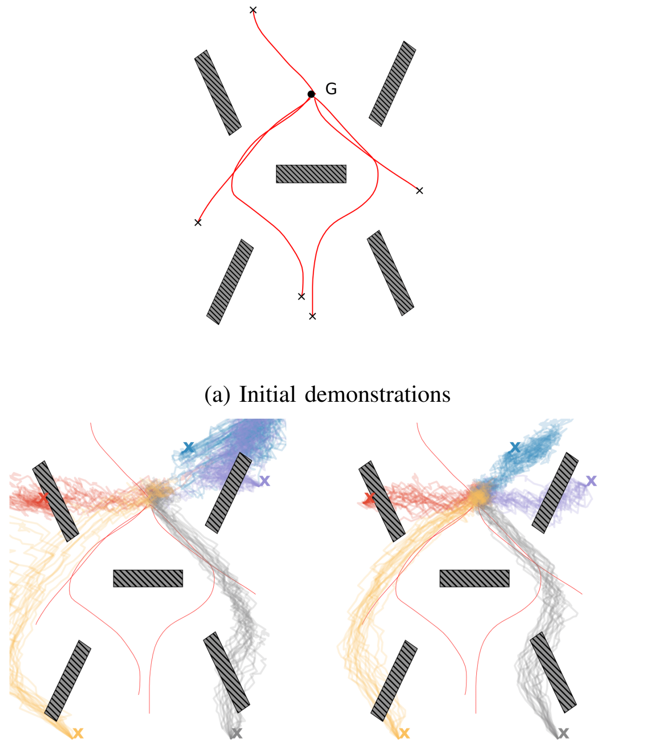
Control policies are defined as the probability distribution of control commands or actions $\boldsymbol{u}$ given the state $\boldsymbol{x}$, denoted as $p(\boldsymbol{u}|\boldsymbol{x})$. They encode the demonstration trajectories along with the dynamics information of the controlled system. As described in Section III-A, we impose a BGMM model structure for the control policy and estimate the parameters of the predictive conditional distribution from the demonstrations.

In this section, we present the proposed active learning of control policies approach. First, a cost function is defined using the epistemic uncertainties in the BGMM control policy and optimized while considering a soft constraint to be on the desired region of the state-space. The robot then asks for a new demonstration around the query point found by the optimization process. The data of the new demonstration is added to the previous dataset and the BGMM parameters are updated. The robot iterates this process until it reaches a predefined percentage of uncertainty reduction.

In order to build the active learning cost function, the covariance matrices of the control policy must be decomposed into its aleatoric and epistemic parts (Section IV-A). Then, we deploy Rényi entropy to calculate epistemic uncertainties in closed-form (Section IV-B). The complete formulation of the resulting cost is presented in Section IV-C.

A. Uncertainty decomposition

The uncertainty in the posterior distribution of the BGMM model encodes the variations in the demonstrations, called aleatoric uncertainty, along with the epistemic uncertainty,



(a) Initial demonstrations

(b) Policy samples from BGMM (c) Policy samples from PoE

Fig. 2: *(a)* Demonstrations and *(b)-(c)* reproductions of a reaching task in a cluttered environment. The goal position is denoted by G and the obstacles are represented as dashed rectangles. The demonstrated trajectories are depicted with red lines. The policy samples acquired from the BGMM and PoE are depicted by colored lines.

measuring the lack of knowledge of the model. These different uncertainty modalities are depicted in Fig. 3 for our illustrative example. In active learning, we are interested in increasing the knowledge of the model, by providing demonstrations around interesting regions of the input space.

In BGMM model, the covariance matrix of the conditional posterior predictive distribution of (11) can be decomposed into aleatoric and epistemic parts as

$$\boldsymbol{L}_k^{o|i} = \boldsymbol{L}_k^{\text{al}} + \boldsymbol{L}_k^{\text{ep}}, \tag{12}$$

where

$$\boldsymbol{L}_k^{\text{al}} = \frac{\nu_k}{\nu_k^{o|i}}\boldsymbol{L}_s, \tag{13}$$

$$\boldsymbol{L}_k^{\text{ep}} = \frac{(\hat{\boldsymbol{x}}^i - \boldsymbol{m}_k^i)^\top \boldsymbol{L}_k^{ii^{-1}}(\hat{\boldsymbol{x}}^i - \boldsymbol{m}_k^i)}{\nu_k^{o|i}}\boldsymbol{L}_s. \tag{14}$$

Notice that the aleatoric uncertainty does not depend on the input point $\hat{\boldsymbol{x}}^i$, while the epistemic uncertainty is a quadratic function of $\hat{\boldsymbol{x}}^i$. The former represents the variability and the noise in the demonstrations and the latter encodes the uncertainty caused by finite data. In robotics, both types of uncertainty are important to capture, i.e. the variations of the demonstrations and the uncertainty in the model, for applications such as compliance adaptation and active learning.

B. *Rényi entropy of the posterior distribution*

When the posterior distribution $p(\boldsymbol{u}|\boldsymbol{x})$ is a multivariate GMM (or can be approximated by one), the information-theoretical Shannon entropy does not admit an analytical form. In order to avoid a significant amount of computational burden for the minimization of active learning cost, we use instead the quadratic Rényi entropy, which admits a differentiable closed form for GMMs [18]. Another reason is that it is very close to Shannon entropy value as will be detailed below.

A random variable $\boldsymbol{U}$ from a multivariate t-distribution $\boldsymbol{U} \sim t_\nu(\boldsymbol{u}|\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\Sigma}(\boldsymbol{x}))$ can be approximated by a multivariate normal distribution with mean $\tilde{\boldsymbol{\mu}}(\boldsymbol{x})$ and covariance $\tilde{\boldsymbol{\Sigma}}(\boldsymbol{x})$ using moment-matching method, so that

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{x}) = \boldsymbol{\mu}(\boldsymbol{x}), \qquad \tilde{\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{\nu}{\nu - 2}\boldsymbol{\Sigma}(\boldsymbol{x}).$$

This approximation can be extended to mixtures using the same mixing coefficients. The Rényi entropy of order α is defined as $H_\alpha(p) = \frac{1}{1-\alpha}\log\int p^\alpha(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ with $\alpha > 0$ and $\alpha \neq 1$. In the limit case where $\alpha \rightarrow 1$, the Rényi entropy is equivalent to the Shannon entropy defined as $H_\alpha(p) = -\int p(\boldsymbol{x})\log p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$. In this paper, we propose to use quadratic Rényi entropy defined as

$$H_2(p(\boldsymbol{u}|\boldsymbol{x})) = -\log\int p^2(\boldsymbol{u}|\boldsymbol{x})\mathrm{d}\boldsymbol{u},$$

since it admits a closed-form expression for GMMs. Note that the Rényi entropy is a non-increasing function of α, so that $H_1(\cdot) > H_2(\cdot)$. In an active learning framework, the entropy can be used as an uncertainty measure to minimize by searching for the queries that have high entropy values. Even though the Shannon entropy is usually used in information theory, maximizing the quadratic Rényi entropy is equivalent to maximizing a lower bound of the Shannon entropy, which would also maximize it suboptimally. The quadratic Rényi entropy for a posterior distribution represented as a GMM $p(\boldsymbol{u}|\boldsymbol{x}) = \sum_{k=1}^{K}\pi_k(\boldsymbol{x})\mathcal{N}(\boldsymbol{\mu}_k(\boldsymbol{x}), \boldsymbol{\Sigma}_k(\boldsymbol{x}))$ can be expressed as [18]

$$H_2(p(\boldsymbol{u}|\boldsymbol{x})) = -\log\sum_{i=1}^{K}\sum_{j=1}^{K}\pi_i(\boldsymbol{x})\pi_j(\boldsymbol{x})e^{\Delta_{ij}(\boldsymbol{x})}, \tag{15}$$

where

$$\Delta_{ij} = \frac{1}{2}\Big(\boldsymbol{\mu}_{ij}\boldsymbol{\Sigma}_{ij}^{-1}\boldsymbol{\mu}_{ij} - (\boldsymbol{\mu}_i^\top\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i + \boldsymbol{\mu}_j^\top\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j) - \log\frac{|\boldsymbol{\Sigma}_i^{-1} + \boldsymbol{\Sigma}_j^{-1}|}{|\boldsymbol{\Sigma}_i^{-1}||\boldsymbol{\Sigma}_j^{-1}|} - d\ \log 2\pi\Big) \tag{16}$$

for the i^{th} and j^{th} components of a GMM, with $\boldsymbol{\Sigma}_{ij} = (\boldsymbol{\Sigma}_i^{-1} + \boldsymbol{\Sigma}_j^{-1})^{-1}$ and $\boldsymbol{\mu}_{ij} = \boldsymbol{\Sigma}_{ij}(\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i + \boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j)$.

Fig. 3 depicts *(a)* the total, *(b)* aleatoric and *(c)* epistemic uncertainties computed via the quadratic Rényi entropy of the BGMM model of our illustrative example (Fig. 2a). Yellow and purple colors depict high and low uncertainties, respectively. Note that the uncertainty of the aleatoric model stays constant as we move away from known data, while it increases in epistemic model. As the epistemic model describes unseen regions, it must be used for an efficient search in the state-space.

C. Information-density cost for active learning

Following a similar approach to information weighted technique described in Section II, we constrain the optimization space by adding a similarity function that measures the closeness to a region of space where we want to improve our model. In this work, we represent this region as a probabilistic density function (pdf). Note that, even though the region of interest may often be represented as a uniform distribution, one may want to favor some parts of this region compared to others using other distributions. Therefore, we can solve the following optimization problem

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} -H_2(p(\boldsymbol{u}|\boldsymbol{x}))-\beta \log p_{\text{sim}}(\boldsymbol{x}), \tag{17}$$

with the epistemic cost in closed-form, to find the next query point $\boldsymbol{x}$, where β is a variable weighing the relative importance of the costs. In practice, uniform distributions will result in negative infinity log probabilities in the outside regions and will not have a defined gradient at the border. Therefore, we approximate the uniform distribution by an MVN using the same mean and diagonal covariance matrix to alleviate this issue. Another problem with the optimization of Eq. (17) is the existence of flat regions from which the optimization cannot escape. To overcome this problem, we propose to approximate the epistemic cost in Eq. (17) as a GMM with a variational distribution $q(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})$ to represent all the regions where epistemic uncertainty is high, using reverse KL divergence as in

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}\, KL\Big(q(\boldsymbol{x})||H_2(p(\boldsymbol{u}|\boldsymbol{x}))+\beta \log p_{\text{sim}}(\boldsymbol{x})\Big). \tag{18}$$

Note that one can also augment the epistemic cost defined in Eq. (17) with other costs (see robotic experiment in Section V.B.), so that $q(\boldsymbol{x})$ can represent a more constrained space (e.g. being away from an undesirable region). We can obtain the next query point either by sampling from $q(\boldsymbol{x})$ or by taking the mean of one of the components. As we add more demonstrations and improve our model using this query point, the optimization in Eq. (18) can be initialized with the parameters of the previous $q(\boldsymbol{x})$, which would increase convergence speed. We expect a decrease of entropy in $q(\boldsymbol{x})$ at each iteration of active learning. This gives us a natural way of monitoring the uncertainty reduction.

Fig. 3*d* shows the information density colormap favoring to be inside of the figure frame where we want to generalize our model. It also shows the GMM contour ellipses (with 1 standard deviation) which approximate the high information-density regions (yellow). The transparency reflects the mixing coefficient of the GMM. We can observe that the highly uncertain regions are well approximated.

V. EXPERIMENTS

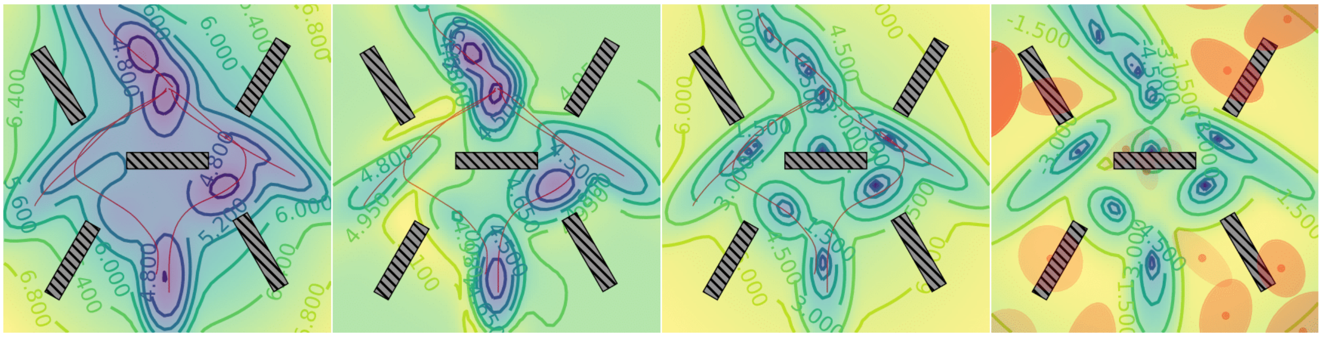
A. Illustrative reaching task

We use the proposed active learning framework to gather iteratively 10 more demonstrations for our illustrative 2D reaching task. At each step, the model informs the teacher on the next query point, given by the mean of the GMM component with the highest mixing coefficient (corresponding to the highest uncertainty). As any sample from that component can be used as a next query point, the closest feasible position to the mean can be chosen if the mean does not correspond to a feasible location, e.g. if it collides with the obstacles.

Note that we are interested in reducing the epistemic uncertainties in the conditional model, which is a function of the input point as in (14). In order to define an entropy reduction, we need a measure that does not depend on the input point. We can thus measure how much the entropy changes via the GMM model which approximates highly uncertain regions. Fig. 4*a-(top)* shows the evolution of the quadratic Rényi entropy of the GMM model across the active learning iterations. Red crosses show the current entropy values, whereas the black curve is 2D polynomial fit to these values. We can observe that the entropy of the GMM is reduced until there is no component left which can specialize in certain regions with small covariance (small covariance means low entropy). After 6 iterations, the entropy starts to increase as the components are more diffused with bigger covariance matrices. We generally observed that the entropy of the GMM behaves similarly to the black curve in Fig. 4*a-(top)*. The evolution of the entropy of the marginal model $p(\boldsymbol{x})$ is represented in Fig. 4*b-(top)*. As expected, the entropy of the marginal model decreases with the quantity of data. Therefore, it results in no explicit method to infer the convergence of the learning process. In contrast, with our GMM model, one can argue that the system has learned a significant percentage of the unseen regions after 6 iterations.

We conducted 5 more experiments performing active learning where new random demonstrations are provided for 5 iterations. The mean and standard deviation of 5 experiments at each iteration are shown in Fig. 4*a-(bottom)* for the GMM model and in Fig. 4*b-(bottom)* for the marginal model. This demonstrates that the random exploration is not guaranteed to reduce the epistemic uncertainties, even in the marginal model.

The resulting reproductions from the chosen random initial test positions using samples from the updated BGMM and PoE policies are shown in Fig. 5a and Fig. 5b, respectively. We observe that both policies successfully avoid all the obstacles in the average, while using PoE framework results in a more stable system. The query points of each iteration of active learning are also labelled in Fig. 5a. We observe that these query points are rather intuitive, as they correspond to locations that could be chosen by a human to better teach the task to the robot. In contrast, informative query points may be very difficult to choose in other cases where the query space is not easily interpretable.



(a) Total (b) Aleatoric (c) Epistemic (d) Information

Fig. 3: Uncertainty colormaps of the learned control policy for a reaching task in a cluttered environment. *(a)*, *(b)* and *(c)* show the total, aleatoric and epistemic uncertainties of the BGMM, respectively. High to low uncertainties are depicted by colors ranging from yellow to purple. *(d)* depicts the information-density cost and the Gaussian components of the GMM model approximating this cost.

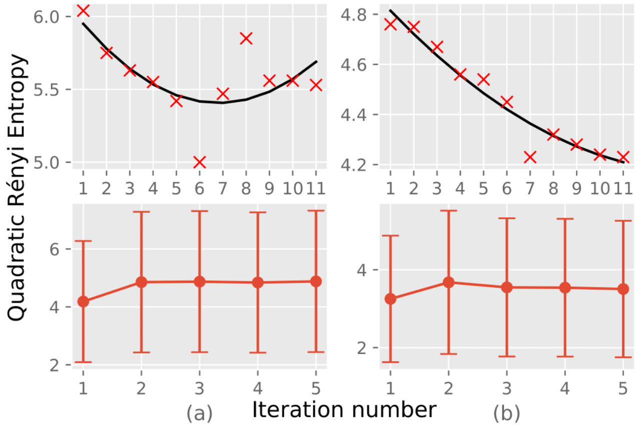
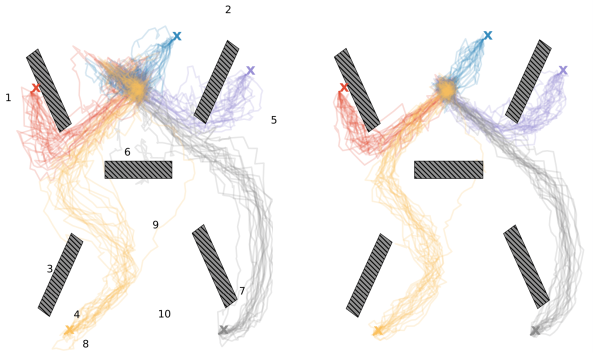


Fig. 4: Evolution of the quadratic Rényi entropy of (a) the GMM model that approximates highly uncertain regions and (b) the marginal BGMM model. Top figures represent the evolution for the proposed active learning, while the error bars in bottom figures show the mean and the standard deviation of 5 different random exploration for 5 iterations.



(a) Policy samples from BGMM (b) Policy samples from PoE

Fig. 5: Reproductions of the learned policy after 10 iterations of active learning. The numbers on (a) denotes the location of the query point at each iteration of active learning.

B. *Robot Experiment*

We investigate the reaching in a cluttered environment task shown in Figure 1 within our active learning framework. The main challenge of this task is to place the cup inside the white box without colliding with the environment and without pouring the cup. The robot can place the cup from any open side of the box, as long as the cup is inside. Planning methods can be applied to find a joint configuration trajectory starting from a given initial configuration of the robot without colliding with the environment, given the size and positions of the obstacles. However, learning control policies using BGMM offers the advantage of sampling the next state much faster than standard planning methods. It also provides a formal way of improving the planned trajectory using active learning framework proposed in this paper. For the improvement of the learned policy, it is difficult for the teacher to choose informative joint configurations intuitively as the demonstrations can take place starting from many different end-effector positions, which correspond to many more joint configurations. Our goal in this experiment is to show that our method provides “intuitive” and informative query points in the joint space of the robot.

We first demonstrate the reaching task from 11 different initial configurations and learn our control policy. Note that the demonstrations are taken from each side of the box, where it was easy to perform kinesthetic teaching. The initial configurations of the demonstrations are depicted in Figure 6 (left).

To improve the model, one need to start from a rather different and informative initial configuration of the arm, which is not easy. Note that the robot has to maintain upright position of the cup to place it inside the box without pouring it and without colliding with the environment. That is why the search space we are interested in is constrained such that we add 2 more cost functions to Eq. (17) in the form of probability distributions: *i*) a cost to keep orientation with respect to x-y axis of the robot base fixed and *ii*) a cost to be within the joint limit range of the robot as

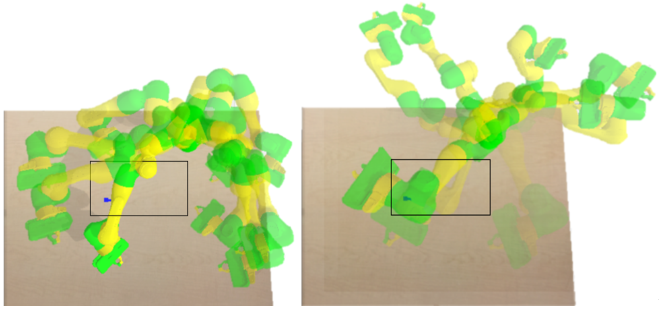


Fig. 6: (Left) Initial configurations of the demonstrations, (Right) Requested initial configurations for demonstration

$$p(\boldsymbol{x}) = H_2(p(\boldsymbol{u}|\boldsymbol{x})) + \beta \log p_{\text{limits}}(\boldsymbol{x}) + \alpha \log p_{\text{upright}}(\boldsymbol{x})$$

where

$$p_{\text{limits}}(\boldsymbol{x}) =$$
$$p_{\text{upright}}(\boldsymbol{x}) =$$

We approximate this cost by a GMM of 10 components minimizing KL divergence between $q(\boldsymbol{x})$ and $q(\boldsymbol{x})$ as in Eq. 18. The resulting query configurations (means of the GMM components) are given in Figure 6 (right). We can see that our GMM could in fact approximate highly uncertain and unseen configurations of the robot as it requests demonstrations around these regions. These configurations are also within the joints range of the robot, and maintain a fixed x-y axis orientation so that the robot will keep the cup upright, without pouring. Although showing that the usefulness of encoding aleatoric uncertainties here is out of scope of this paper, it has been exploited in the previous work in [9]. Since the aperture size of the sides are big enough, one can imagine exploiting high variations in the demonstrations while the end-effector enters one side of the box. The learned model would create compliant control commands in these areas which would help the teacher to correct the robot movement during a failing execution. Note that GPR could not encode aleatoric uncertainties.

VI. Conclusion

This paper presented a novel active learning framework allowing a robot to ask for informative new demonstrations. The presented framework is based on an information-density cost built from a representation of the epistemic uncertainties of a BGMM model. A closed-form cost solution can be obtained thanks to the properties of the quadratic Rényi entropy, which admits a closed-form for GMMs. New query points can then be efficiently obtained by maximizing a GMM approximation of the proposed active learning cost. Our experiments showcase that our approach allows a robot to improve its representation of a task, as well as its corresponding generalization capabilities.

The model in our work can assess the uncertainty of the control command given the current state. However, in many application in robotics, it is necessary to propagate these uncertainties to determine the uncertainty on the whole trajectory. Future work should focus on either how to propagate uncertainties in the state-action policies, or using a model that can already reason about the uncertainty of the trajectory. Another future work consists in extending our results to theoretically determine a threshold to stop the learning process, which in turn would be useful for determining a sufficient number of demonstrations so that the model can generalize the fastest in the desired space. We believe that the framework can then be used to answer two of the main questions of LfD, which are *i*) Where to give demonstrations? and *i*) How many demonstrations are required?.

References

[1] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," in *Lazy learning*. Springer, 1997, pp. 75–113.

[2] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2002, pp. 1547–1554.

[3] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems (NIPS)*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. USA: Curran Associates, Inc., 2013, pp. 2616–2624.

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

[5] T. Cederborg, L. Ming, A. Baranes, and P.-Y. Oudeyer, "Incremental local online Gaussian mixture regression for imitation learning of multiple tasks," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.

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

[7] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[9] E. Pignat and S. Calinon, "Bayesian Gaussian mixture model for robotic policy imitation," *IEEE Robotics and Automation Letters*, Oct 2019.

[10] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[11] M. P. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics," *Found. Trends Robot*, vol. 2, pp. 1–142, Aug. 2013.

[12] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[13] 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.

[14] 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.

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

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

[17] I. Abraham and T. D. Murphey, "Active learning of dynamics for data-driven control using Koopman operators," *IEEE Transactions on Robotics*, 2019.

[18] F. Nielsen, "Closed-form information-theoretic divergences for statistical mixtures," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 1723–1726.