EPFL

Thèse n°9368

Learning strategies and representations for intuitive robot learning from demonstration

Présentée le 17 décembre 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Thibaut Antoine KULAK

Acceptée sur proposition du jury

Prof. P. Frossard, président du jury
Dr J.-M. Odobez, Dr S. Calinon, directeurs de thèse
Dr D. Losey, rapporteur
Dr S. M. Nguyen, rapporteuse
Dr L. Ott, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

Acknowledgements

First I would like to thank my supervisor Sylvain for advising me throughout the thesis, giving me the freedom to pursue my ideas and trusting me. I am also very grateful to my thesis director Jean-Marc for his advice during these years and to the jury members Prof. Frossard, Dr Nguyen, Dr Losey, Dr Ott for accepting to review this thesis and be part of the oral exam.
I would not have completed this thesis without the continuous support of my girlfriend Flore, who gave me endless love without which I could not have overcome the many difficulties and obstacles a PhD entails. I am very grateful to my family, my parents for their continuous support during my life and my studies, no matter the choices I made.
A PhD is a long delicate process on which one name appears, but it cannot be made alone and I would like to thank the many people that made it possible. My colleagues and friends Christian, Olivia, Hakan, Emmanuel, João, Martin, Nicolas and many others for the many coffees and discussions that made the work definitely more enjoyable, especially in harder times when the research direction was not so clear for me and lots of uncertainties and doubts troubled me. Special thanks to my friends and climbing partners, Alex, Ludo for their never-ending happiness that always recharged motivation, and to all of my friends. I would also like to thank all of the people that contributed in making me feel good in this new environment and country I moved in, notably the people from the AAGB association in my village. Feeling at home and welcome after a drastic life change definitely contributed to my going through this PhD. Finally, thanks to Pimousse for bringing softness into the last months of this journey.

Les Granges, 30th August 2021 T. K.

Abstract

Robots are becoming more and more present around us, both in industries and in our homes. One key capability of robots is their adaptability to various situations that might appear in the real world. Robot skill learning is therefore a crucial aspect of robotics aiming to provide robots with programs enabling them to perform one or several tasks successfully. While such programming is usually done by an engineer or a developer, making robot programming available to anyone would dramatically increase the range of applications currently feasible for robots. Learning from Demonstration (LfD) is a robot skill learning paradigm addressing this aim by developing intuitive frameworks for non-expert users to easily (re)program robots.
While Learning from Demonstration has emerged as a successful way to program robots, several limitations remain to be addressed. Typical approaches still require some forms of preprocessing, such as the alignment of the demonstrations, or the choice of the movement representation. Also, the algorithms have to run with a relatively low number of demonstrations that human users are typically willing to give, while being performant, adaptable and generalizable to new situations. In this thesis, we propose to address these shortcomings with methods that make Learning from Demonstration more intuitive and user-friendly. We notably propose a novel movement representation requiring no demonstration alignment, and active learning strategies that permit to learn complex skills from fewer demonstrations.

Keywords: Learning from demonstrations, active learning, robot learning, imitation learning

Résumé

Les robots occupent une place de plus en plus importante autour de nous, à la fois pour des applications industrielles et domestiques. Ils doivent pouvoir s'adapter à la diversité des situations qui peuvent apparaître dans leur environnement, ce qui nécessite une bonne programmation robotique. Celle-ci est cruciale car elle fournit aux robots des programmes leur permettant de réaliser une ou plusieurs tâches avec succès, et elle est habituellement effectuée par un ingénieur ou un développeur spécialisé. Nous pensons qu'ouvrir les portes de la programmation robotique à des utilisateurs non experts élargirait drastiquement le champ des applications robotiques possibles. L'apprentissage par démonstration est une des voies possibles pour atteindre cet objectif, il ne nécessite pas de connaissances en programmation de la part de l'utilisateur puisque ce dernier programme le robot simplement en lui montrant comment faire la tâche en question.
Bien que l'apprentissage par démonstration ait été appliqué avec succès pour programmer des robots, un certain nombre de questions restent ouvertes et limitent le champ d'applications. Les approches existantes nécessitent généralement un prétraitement des démonstrations, tel qu'un alignement, ainsi qu'un choix approprié de la représentation du mouvement. D'autre part, les algorithmes développés doivent fonctionner avec un nombre relativement restreint de démonstrations qu'un utilisateur est prêt à effectuer. Enfin, ils doivent être performants, s'adapter et généraliser à des nouvelles situations. Dans cette thèse, nous proposons des solutions à ces limitations avec des méthodes qui rendent l'apprentissage par démonstration plus intuitif et facile d'utilisation. Notamment, nous proposons une nouvelle manière de représenter les mouvements qui présente l'avantage de ne pas nécessiter de prétraitement, ainsi que des méthodes d'apprentissage actif qui permettent d'apprendre des tâches complexes avec un nombre restreint de démonstrations.

Mots clefs : Apprentissage par démonstration, apprentissage actif, apprentissage robot, apprentissage par imitation

Contents

Acknowledgements **3**

Abstract (English/Français) **5**

List of figures **13**

List of tables **15**

1 Introduction **17**
1.1 ROSALIS project 18
1.2 Thesis organization 19

2 Background **21**
2.1 Robot skill learning 21
2.1.1 Imitation learning 21
2.1.2 Reinforcement learning 22
2.1.3 Intrinsically-motivated learning 23
2.2 Movement primitives 23
2.3 Learning distributions 26

3 Fourier Movement Primitives: an approach for learning robot skills from demonstrations **31**
3.1 Introduction 31
3.2 Related work 33
3.2.1 Dynamical-system-based approaches 33
3.2.2 Probabilistic approaches 34
3.2.3 Constraint-based approaches 34
3.3 Preliminaries 35
3.3.1 Discrete Fourier transform 35
3.3.2 Inverse discrete Fourier transform 36
3.4 Fourier movement primitives 36
3.4.1 Imitation learning 36
3.4.2 Mapping partial trajectories to Fourier domain 38
3.4.3 Tracking in the Fourier domain 40

3.4.4 Multidimensional case . . . 41
3.5 Experiments . . . 42
3.5.1 Data acquisition and preprocessing . . . 42
3.5.2 Polishing task . . . 42
3.5.3 8-shape drawing . . . 45
3.5.4 Real-world wiping task . . . 47
3.6 Discussion . . . 49
3.7 Conclusion . . . 50

4 Active Learning of Bayesian Probabilistic Movement Primitives 53
4.1 Introduction . . . 53
4.2 Related work . . . 55
4.3 Bayesian ProMPs . . . 56
4.3.1 Contextual ProMP . . . 56
4.3.2 Problem formulation . . . 57
4.3.3 Bayesian Gaussian Mixture Model (BGMM) . . . 57
4.4 Active learning of ProMPs . . . 59
4.4.1 Uncertainty decomposition . . . 59
4.4.2 Uncertainty measurement . . . 59
4.5 Illustrative examples . . . 60
4.5.1 Why epistemic uncertainties? . . . 61
4.5.2 Visualization of uncertainties . . . 62
4.5.3 Why not Gaussian Processes? . . . 65
4.6 Experiments . . . 65
4.6.1 Simulated pouring . . . 66
4.6.2 Real robot pouring task . . . 71
4.7 Conclusion . . . 72

5 Combining Social and Intrinsically-Motivated Learning for Multi-Task Robot Skill Acquisition 73
5.1 Introduction . . . 73
5.2 Related Work . . . 75
5.3 Bayesian Movement Representation . . . 76
5.4 Active learning modalities . . . 77
5.4.1 Manipulation task . . . 77
5.4.2 Imitation learning . . . 77
5.4.3 Intrinsically-motivated learning . . . 78
5.4.4 Choosing the learning modality . . . 79
5.5 Experiments . . . 80
5.5.1 Waste throwing task . . . 81
5.5.2 Imitation learning . . . 83
5.5.3 Intrinsically-motivated learning . . . 85
5.5.4 Choice of learning modality . . . 86

5.6 Conclusion 89

6 Summary and future work **91**
6.1 Possible research directions 92
6.1.1 Fourier movement primitives for discrete motions 92
6.1.2 Quantifying the number of demonstrations required 93
6.1.3 Considering additional learning modalities 93
6.1.4 Model-based learning approaches 95

Bibliography **107**

Curriculum Vitae **109**

List of Figures

3.1 Rythmic tasks such as wiping need to be demonstrated in few demonstrations (**top**), while the robot should extract the important motion features (e.g. amplitude, frequency and phase) and generalize it in a consistent, safe manner (**bottom**). 32

3.2 Illustration of reconstructed signals with one Fourier basis function (for $k = 3$). The orange and blue points have the same amplitude but not the same phase, which results in the same signals that are shifted in time. 35

3.3 Illustration of the advantage of Fourier basis functions for non-aligned point-to-point demonstrations 39

3.4 Demonstrations of the polishing task. 43

3.5 Polishing from different initial positions with FMP. 43

3.6 Heatmaps of distribution learned with ProMP-VM. 44

3.7 Conditioning on initial position with ProMP-VM. 45

3.8 Samples of the learned 8-shape distribution. 46

3.9 Heatmaps of the learned 8-shape distribution. 46

3.10 8-shape from different initial positions with FMP. 48

3.11 Generated trajectories of length 400 (20s) for a given initial position. 49

4.1 Overview of the pouring task with a 7-axis robot. 54

4.2 Probability density functions of univariate t-distributions for several degrees of freedom. 58

4.3 Visualization of the aleatoric and epistemic uncertainties of a Bayesian Gaussian mixture model 64

4.4 Visualization of the uncertainties of a Gaussian Process 66

4.5 Overview of the simulated pouring environment. 67

4.6 Subset of demonstrations for different contexts. 68

4.7 Quantitative results for simulated 1D context pouring. 68

4.8 Quantitative results for simulated 2D context pouring. 70

4.9 Quantitative results for simulated 3D context pouring. 71

4.10 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. 71

5.1 Object trajectory for 6 demonstrations of the database (3 dynamic demonstrations in orange, and 3 non-dynamic demonstrations in blue). 82
5.2 Desired final object positions. The grey rectangle represents the goal space $\mathcal{G}$. Blue/orange dots show the final object position of respectively the non-dynamic/dynamic demonstrations of the database. 83
5.3 Evolution of the active imitation learning strategy. The goal space is represented in this figure. Grey stars represent the final object position of the available demonstrations, and orange stars the selected goal to query. The transparent ellipses show the marginal distribution of the BGMM on the goal space. 84
5.4 Evaluation of imitation learning strategy. 85
5.5 Influence of demonstrations for intrinsically-motivated learning strategy. 86
5.6 Evaluation of intrinsically-motivated learning strategy (task cost in logarithmic scale). 87
5.7 Example of a learning process in which the learning strategy is selected at each step based on the proposed active learning method. 87
5.8 Evaluation for the choice of the learning strategy. (**Blue**: Random choice between Imitation Learning (IL) and Intrinsically-motivated Learning (IML), with probability 36% of choosing IML. **Orange**: Active choice of the learning modality. **Green**: Selecting always imitation learning. **Red**: Selecting always imitation learning, but using the same number of demonstrations as the active arbitration strategy (Orange boxplot). **Purple**: Selecting always intrinsically-motivated learning modality.) 88

List of Tables

3.1 Quantitative comparison of distributions learned with ProMP-VM-Mult and FMP versus ground truth distribution for 8-shape task. 47

1 Introduction

Robots are becoming ubiquitous in our society, as they are used in many areas such as manufacture, warehouses, logistics or agriculture. However, the difficulty to program them is a limiting factor as robot programming typically requires many hours of work of a dedicated engineer for a given task. Democratization of robotics is still in its early stages, which hinders the potential of robotic applications. Let us make a comparison: back in the 90s, creating a website was hard and only accessible to a small group of technical experts. Now that we have efficient tools allowing anyone without specific knowledge to create his/her own website, we have dramatically increased the business potential and creativity of the web, and most of today's websites would not exist if such tools would not have been created. Similarly, we believe that efficient tools permitting any user to intuitively and easily program robots would open the way to a huge diversity of new robotic applications [68]. For instance, robot programming would be possible for small companies which do not have the financial resources to support a dedicated robot engineer [103], or even for private individuals for domestic applications. We simply cannot imagine all the fallouts such tools would have on our society [120], likely improving the lives of billions of people [47, 101].

In this thesis, we support this long-term goal of developing tools making robot programming accessible to anyone, hence democratizing robotics. A popular framework for it is the Learning from Demonstrations framework [22, 115, 3], by which human users can teach robot tasks by providing them with demonstrations, i.e., showing the robot how to do the task. But this framework suffers from several limitations hingering its applications potential. Notably, it may not be trivial for a human user to provide good informative demonstrations, as he/she does not know the processus by which the robot learns [25]. Also, human users are only willing to provide a handful of demonstrations, after which they become bored and do not wish to use the system. In this thesis, we try to address these challenges by proposing methods that make Learning from Demonstrations more intuitive and user-friendly, hence taking a step towards robotics democratization.

1.1 ROSALIS project

This thesis is part of the ROSALIS project (Robot skills acquisition through active learning and social interaction strategies), which proposes to rely on natural interactions for robot skill learning. Most efforts in robot learning from demonstration are turned toward developing algorithms for the acquisition of specific skills from training data. While such developments are important, they often do not take into account the social structure of the process, in particular, that the interaction with the user and the selection of the different interaction steps can directly influence the quality of the collected data. Similarly, while skills acquisition encompasses a wide range of social and self-refinement learning strategies, including mimicking (without understanding the objective), goal-level emulation (discovering the objectives by discarding the specific way in which a task is achieved), exploration with self-assessed rewards or feedback from the users, they each require the design of dedicated algorithms, but the ways in which they can be organized have been overlooked so far. In ROSALIS, we propose to rely on natural interactions for skill learning. Active learning methodologies will be developed, relying on heterogeneous sources of information. We target applications of robots in both manufacturing and home/office environments, both requiring re-programming in an efficient and personalized manner.

1.2 Thesis organization

Background This chapter introduces the research background of the thesis. In Section 2.1, an overview of the different existing methods for robot skill learning is presented, with a focus on methods that are based on human demonstrations. In Section 2.2, we discuss the need for learning a probabilistic representation of movement primitives, and present the LfD framework of probabilistic movement primitives (ProMPs). In Section 2.3, we present the machine learning models for approximating probability distributions that will be used in the thesis, namely the frequentist and Bayesian versions of Gaussian mixture models.

Fourier movement primitives This chapter proposes a LfD approach that leverages the theoretical properties of the Fourier transform to propose a method for learning robot skills from demonstrations that is intuitive and easy to use for users. Notably, it does not require demonstrations to be aligned by the user, nor the user to carefully choose the basis functions that are used by the algorithm to approximate the movements, which is usually the case.

Active learning of Bayesian probabilistic movement primitives This chapter addresses the problem of quantifying what constitues a good demonstration. This is a typical difficulty of LfD methods where the human user does not know how to provide informative demonstrations to the robot. In this chapter, we propose to quantify and leverage the uncertainties of the movement representation for actively requesting demonstrations to the human user. This reduces the human user cognitive load in two ways: he/she does not have to think about the next demonstration to provide, and it reduces the overall number of demonstrations he/she has to give.

Combining social and intrinsically-motivated learning for multi-task robot skill acquisition In this chapter, we extend the framework of the previous chapter to combine several robot learning modalities: not only does the robot learn from demonstrations, but it also has the possibility to learn by itself from experience with intrinsically-motivated learning. We propose a unified framework to combine those learning modalities, as well as an active learning method for choosing between them.

Summary and future work This chapter summarizes the contributions of the thesis, and introduces several open research directions that could be considered for future work.

2 Background

This chapter introduces the research background of the thesis. We start by presenting an overview of the different existing methods for robot skill learning in Section 2.1, with a focus on methods that are based on human demonstrations. In Section 2.2, after discussing the need to learn a probabilistic representation of movement primitives, the LfD framework of Probabilistic Movement Primitives (ProMPs) is presented. In Section 2.3, we present the machine learning models for approximating probability distributions that will be used in the thesis, namely the frequentist and Bayesian versions of Gaussian mixture models.

2.1 Robot skill learning

Robot skill learning encompasses a wide range of techniques and frameworks, the most popular are:

- **imitation learning**, where the robot learns by imitating an expert (human user)
- **reinforcement learning**, where the robot learns by itself thanks to guidance provided by a reward function, that acts as a reward or retribution.
- **intrinsically-motivated / curiosity-driven learning**, where the robot learns by itself, following an internal reward function usually rewarding some form of curiosity.

All frameworks share the same high-level goal: proposing techniques that permit to alleviate the need to manually program each and every robot behavior. We review here the advantages and drawbacks of those frameworks as well as the relevant literature.

2.1.1 Imitation learning

As robots move from simple controlled environments to more complex real-world situations, their programming is becoming more and more challenging and expensive. It might be easier

for a teacher to demonstrate a desired behavior rather than to manually engineer it. This process of learning from demonstrations and the study/design of algorithms doing so is called imitation learning [98]. This is an active area of research with two main trends: some works attempt to learn to replicate the desired behavior directly, a process which is called *behavioral cloning* [8], or *mimicking* [142]. Other works attempt to learn the hidden objectives of the desired behavior from demonstrations, a process which is called *inverse optimal control* [69], *inverse reinforcement learning* [118], or *emulation* [142]. The choice of behavioral cloning versus inverse optimal control is very problem dependent and not easy, and one should consider what is the most parsimonious description of the behavior (policy or reward).

Another important distinction can be made between approaches that model the system dynamics and those that do not. The former are called *model-based* approaches, and the latter *model-free* approaches. In *model-free* approaches, the system dynamics are only implicitly encoded in the policy learned. One possibility is to directly learn the mapping from state to action (the policy) from the demonstrations [70], but this can cause stability and safety problems as the learned policy is applied on the real robot. But in many robotic systems, position/velocity/torque controllers are available and one can assume that the system is fully actuated. This notably permits to learn skills at the level of the trajectory instead of the control commands level, and this has been widely and successfully used in many approaches [14, 21, 105, 1]. In the absence of such controllers (i.e., in unknown dynamics), it is considerably more difficult to use *model-free* approaches at the trajectory level, although some recent approaches have proposed possible ways [48, 57]. *Model-based* approaches alleviate this by explicitly modeling the system dynamics, and leveraging it for learning a policy [40] or for reward learning [32]. However, it is often challenging to learn the system dynamics of a real robotic system, for instance for tasks involving contacts.

2.1.2 Reinforcement learning

Reinforcement learning is a popular skill learning framework by which an agent learns to maximize an external signal coming from its environment (the reward) [134]. It is a very active research area of artificial intelligence, that has proven very successful for achieving super-human performance for, e.g., Atari games [91] and the Go game [132], thanks to its combination with deep learning (i.e., deep reinforcement learning). Initially designed for discrete state-action spaces, deep reinforcement learning has been extended successfully to continuous state-action spaces [83]. The main drawback of this framework is the usually very high number of trials required to learn a task (for instance, in a recent work an equivalent of 3000 hours of robot interaction time are needed to learn hand-eye coordination for robotic grasping [82]). Though methods have been proposed to alleviate this limitation and make real-world robotic applications possible [46, 28, 135], this is an active area of research that remains unsolved. The subfield of reinforcement learning that has proven the most successful on robotic applications is policy search [35], which focuses on finding good parameters for a given policy parametrization. It usually requires fewer data and can cope with high-dimensional state and action spaces, hence

being better suited to robotic applications [36, 74, 81]. There remains one significant limitation of this framework: the need to carefully choose the reward function. It is known that the task success is highly dependent on the appropriate choice of the reward signal [93]. Choosing the reward signal carefully requires great knowledge and experience, and is therefore non-trivial for non-expert users. Providing binary reward signals [140] is a potential solution to circumvent this limitation, but it complexifies the whole problem as the reward signal does not provide guidance anymore, and it therefore requires more robot trials, or another form of guidance (e.g., human demonstrations).

2.1.3 Intrinsically-motivated learning

Intrinsically-motivated learning (a.k.a. curiosity-driven learning, self-paced learning, self-supervised learning) is a subfield of reinforcement learning [45] that does not require the careful design of an external reward function. It has emerged as an efficient approach for autonomous lifelong learning in robots [99, 123], and it is inspired by the ability of humans to discover how to produce interesting effects in their environments [141, 33, 12]. In [12], psychologists suggested that exploration might be triggered and rewarded for situations that include novelty/surprise. They observed that the most rewarding situations were those with an intermediate level of novelty, between already familiar and completely new situations. This also seems to be confirmed by recent neuroscience studies showing that dopamine might be released, not only for predicting external rewards such as food, but also for internal rewards such as prediction errors [60]. This suggests that intrinsic motivation systems might be present in the brain, potentially by the presence of signals related to prediction errors. Given this background, a way to implement an intrinsic motivation system might be to build a mechanism which can evaluate the degree of novelty of different situations from the point of view of a learning robot, and then designing an associated reward being maximal when these features are in an intermediate level. Maximizing this reward can then create an active exploratory behavior [89, 99, 100]. Curiosity-driven learning has been successfully used for robot skill acquisition [122], for instance for learning tactile skills [104], motion planning [49], learning action sequences for high-dimensional video inputs [76], or learning to manipulate wooden blocks [94].

2.2 Movement primitives

A main difficulty in robot skill learning lies in the high dimensionality of the problem. A task usually involves a high-dimensional state space (e.g., 7 degrees of freedom robot joint position and velocity) over a long inherently continuous time horizon. Learning mappings or distributions for such a high-dimensional space is difficult, which is why a popular approach is to use more compact representations of the robot's control policy [6], for example using movement primitives. Modulating the movement primitives parameters permits imitation as well as reinforcement learning, and adaption to different situations. Such formulations have been extensively studied and used for robotic applications such as hitting [66, 73], ball-in-the-cup [74], grasping [139],

throwing [31], and pancake flipping [77].

The aim of a movement primitives framework is a modular control architecture that permits to create complex robot movement out of simple elemental movements. Such framework should provide the following characteristics:

- Parallel activation of movement primitives
- Combination of movement primitives for smooth blending
- Modulation of the movement primitives (e.g., to a desired final position, velocity or via-points)
- Learnable from demonstrations and/or reinforcement learning
- Applicable to stroke-based and periodic-based movements

A popular approach for learning robot movements is to learn a probabilistic representation of the trajectories, either at the level of the movement primitives [23, 107, 105], or directly at the level of the trajectory [20, 23, 24]. This gives the following benefits:

- Adaptation and modulation can be made via conditioning (e.g., conditioning the distribution on the desired final position)
- Composition of movements can be achieved by making a product of probability distributions
- It can encode the variance of the movement, which can for example be used for minimal intervention control
- It models the covariance between trajectories of different degrees of freedom, and can therefore be used to couple the joints of a robot

Probabilistic Movement Primitives (ProMPs) [107, 105] have emerged as a popular tool due to their simplicity, and to their efficient combining of movement primitives and probability distribution learning. The method therefore combines efficiently the above mentionned capabilities of probabilistic trajectory learning as well as the low dimensionality of movement primitives. The main idea of ProMP is to treat a movement as a distribution over trajectories, learned at the level of movement primitives for a more compact representation. We now describe more formally the framework of ProMPs, as it is a tool that we will use throughout the thesis.

Probabilistic movement primitives

Let us denote as $\boldsymbol{y}_t$ the observation at time t of a trajectory. Such observation variable of size D can typically contain all joint angles and velocities. A trajectory (typically, a demonstration) $\boldsymbol{\tau}$ of

size TD is the concatenation of T observations $\boldsymbol{y}_t$:

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{y}_0 \\ \vdots \\ \boldsymbol{y}_T \end{bmatrix}. \tag{2.1}$$

ProMP uses a nD dimensional weight vector $\boldsymbol{w}$ to compactly represent a single trajectory, where n is typically much smaller than T. The probability of observing $\boldsymbol{y}$ at time t is given by a linear basis function model

$$p(\boldsymbol{y}_t|\boldsymbol{w}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{y}_{1,t} \\ \vdots \\ \boldsymbol{y}_{D,t} \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\Phi}_t^\top & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{\Phi}_{t^\top} \end{bmatrix} \boldsymbol{w}, \boldsymbol{\Sigma}_y \right) = \mathcal{N}\left(\boldsymbol{y}_t|\boldsymbol{\Psi}_t^\top \boldsymbol{w}, \boldsymbol{\Sigma}_y\right), \tag{2.2}$$

where $\boldsymbol{\Phi}_t$ defines the n dimensional time-dependent basis matrix function indexed at time t, n denotes the number of basis functions, and $\boldsymbol{\Sigma}_y$ is the $D \times D$ observation noise variance.

A distribution over the weight vector $p(\boldsymbol{w};\theta)$ with parameters θ is introduced to capture the variance of the trajectories. By marginalizing out the weight vector $\boldsymbol{w}$ the trajectory distribution can be computed

$$p(\boldsymbol{\tau};\theta) = \int p(\boldsymbol{\tau}|\boldsymbol{w}) p(\boldsymbol{w};\theta) d\boldsymbol{w}. \tag{2.3}$$

We typically choose a distribution $p(\boldsymbol{w};\theta)$ for which this integral can be computed analytically. The most popular and simple choice is a Gaussian distribution $p(\boldsymbol{w};\theta) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$ over $\boldsymbol{w}$, which yields the following state distribution:

$$\boldsymbol{y}_t = \int \mathcal{N}\left(\boldsymbol{y}_t|\boldsymbol{\Psi}_t \boldsymbol{w}, \boldsymbol{\Sigma}_{\boldsymbol{y}}\right) \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}) d\boldsymbol{w} = \mathcal{N}\left(\boldsymbol{y}_t|\boldsymbol{\Psi}_t \boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Psi}_t^\top \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\Psi}_t + \boldsymbol{\Sigma}_{\boldsymbol{y}}\right). \tag{2.4}$$

Learning the parameters

The parameters $\theta = \{\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w\}$ can be learned from multiple demonstrations $\{\boldsymbol{\tau}_i\}_i$ by maximum likelihood estimation of the parameters $\{\theta, \boldsymbol{\Sigma}_y\}$ of the Hierarchical Bayesian Model (HBM) $p(\boldsymbol{\tau};\theta)$, for example using Expectation Maximization [43]. Most often [44, 105, 108, 88], the HBM is not optimized directly, but a two-stage process is taken to learn the parameters $\{\theta, \boldsymbol{\Sigma}_y\}$:

1. Convert the trajectories $\{\boldsymbol{\tau}_i\}_i$ to their compact representation $\{\boldsymbol{w}_i\}_i$ using linear regression. Eventually deduce (or directly give) Σ_y
2. Learn the parameters θ by maximizing the likelihood $p(\boldsymbol{w};\theta)$

It has been shown in [43] that, in the case where the full trajectories are observed (i.e., no missing data in the demonstrations), this simplified learning procedure gives similar results to the

alternative HBM optimization, which is why it is usually preferred for its simplicity.

Choice of the approximate distribution

While the most simple choice for approximating $p(\boldsymbol{w}; \theta)$ is to use a Gaussian distribution, it has been shown that it might not be sufficient to characterize the variability of demonstrations, and a mixture of Gaussians is sometimes preferred [44].

2.3 Learning distributions

We have presented in previous section a popular framework for learning movement primitives. This framework is probabilistic and aims at representing distributions of trajectories, that can, typically, be used for generalization and adaptation to new situations. In this section, we present several methods for learning distributions that will be used throughout the thesis.

We do not aim at providing a full review of methods for learning distributions, as it is beyond the scope of this thesis, but rather focus on the methods that have been chosen in the thesis. From previous section, we have seen that we have to choose a way to learn the distribution over the weight vectors, as it in turns induces a distribution over the trajectories. The distribution will then be used for conditioning, and hence we choose to learn them with approximate distributions providing analytical conditional distributions. A possible and popular way to do so is to choose a mixture of Gaussians as approximate distributions. In the following subsection, we present the widely-used framework for learning Gaussian mixture models (GMMs) with maximum likelihood. We then present the Bayesian Gaussian mixture model, which is a natural extension that presents interesting properties and alleviates some of the drawbacks of standard maximum likelihood GMMs.

In this section, we suppose we have a dataset of N D-dimensional observations $\{\boldsymbol{x}_1, \dots \boldsymbol{x}_N\}$ that we wish to model. It can be arranged as an $N \times D$ dimensional matrix $\boldsymbol{X}$ in which the n^{th} row is given by $\boldsymbol{x}_n^\top$. Our goal is to learn a joint ditribution $p(\boldsymbol{x})$ over the observation space.

Gaussian mixture models The Gaussian mixture distribution can be written as a linear superposition of Gaussians:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.5}$$

where K is the number of Gaussians, π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are respectively the scalar mixture coefficient, D dimensional mean and $D \times D$ dimensional covariance matrix of the k^{th} Gaussian component of the mixture.

Let us write respectively $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ for the concatenations of the π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ along the first dimension. Assuming that our N observations $\boldsymbol{X}$ are drawn independently from the distribution,

we can express the log likelihood as:

$$\log p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right) \tag{2.6}$$

We aim to find the parameters $\theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ that maximize the likelihood $p(\boldsymbol{X}|\theta)$. Unlike the case of a single Gaussian, the maximum likelihood solution cannot be obtained analytically in the Gaussian mixture case. One way to maximize this likelihood is to use gradient-based techniques [144, 53]. Another (more popular) way is the Expectation Maximization (EM) algorithm [37], which does not require learning rates and automatically enforces all GMM constraints (mixture components summing to 1 and positive definite covariances). The EM algorithm is described in Algorithm 1, for full derivations on how the EM equations are obtained the reader can refer to [15].

It is important to note that the EM algorithm is a local algorithm, and is hence not guaranteed to find the global maximum. The initialization of the parameters at the beginning of EM is therefore very important, and is usually done with the K-Means algorithm [86]. The K-Means algorithm is used to cluster the points into K clusters, the mixture coefficients, the means and the covariance matrices can then respectively be initialized to the fraction of points, the means, and the covariance matrices of each cluster.

Bayesian Gaussian mixture models

In this subsection, we discuss the Bayesian version of Gaussian mixture models, which provides the advantage of recovering not only the most likely set of parameters, but a distribution over the parameters.

As we have seen in previous subsection, the maximum likelihood estimation of Gaussian mixtures solves the following optimization problem

$$\theta = \arg\max_{\theta} \log p(\boldsymbol{X}|\theta). \tag{2.7}$$

Maximum Likelihood is well known for its tendency to overfit data, and for preferring complex models, since they have more parameters and fit the data better. Therefore, maximum likelihood cannot optimize model structure. Bayesian models provides a solution to these problems. Rather than focusing on a single model, it learns a whole class of models. For each model, the posterior probability given the data is computed, and prediction is made by averaging the model predictions weighted by their posterior probabilities. This avoids overfitting, but unfortunately Bayesian models are often intractable [16]. We propose to use *Variational Bayes*, an elegant framework for Bayesian computations in graphical models. This framework permits to have analytical posterior distributions as well as predictive densities, as we will see below. We start by introducing more

Algorithm 1: EM algorithm for Gaussian mixtures

Data: a $N \times D$ dimensional dataset of observations $\boldsymbol{X}$, the number of Gaussians K
Result: GMM parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$

1. Initialize the mixture coefficients $\{\pi_k\}_{k=1}^K$, means $\{\boldsymbol{\mu}_k\}_{k=1}^K$ and covariance matrices $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$, and evaluate the initial log likelihood
2. **Expectation step** Calculate the responsibilities r_{nk} using the current parameters values:
$$r_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$
3. Re-evaluate the parameter values using the current responsibilities:
$$\begin{aligned}\boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k}\sum_{n=1}^N r_{nk}\boldsymbol{x}_n \\ \boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k}\sum_{n=1}^N r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{\text{new}})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{\text{new}})^\top \\ \pi_k^{\text{new}} &= \frac{N_k}{N},\end{aligned}$$
where $N_k = \sum_{n=1}^N r_{nk}$
4. Evaluate the log likelihood
$$\log p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \log\left(\sum_{k=1}^K \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)$$
and check for convergence of the log likelihood, if it has not converged return to step 2.

formally the Bayesian treatment of the Gaussian mixture model.

By Bayes rule we have that:

$$p(\theta|\boldsymbol{X})p(\boldsymbol{X}) = p(\boldsymbol{X}|\theta)p(\theta), \tag{2.8}$$

which implies the following for the posterior distribution:

$$p(\theta|\boldsymbol{X}) \propto p(\boldsymbol{X}|\theta)p(\theta). \tag{2.9}$$

The idea of the Bayesian Gaussian mixture model is to approximate the full posterior distribution $p(\theta|\boldsymbol{X})$ over the parameters θ. Under an appropriate choice of priors, it has been shown that this

can be done using a generalization of the standard EM algorithm [7].

A Dirichlet prior is used for the mixing coefficients and a normal inverse Wishart prior for the means and precision matrices:

$$p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\boldsymbol{\pi})p(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.10}$$

$$p(\boldsymbol{\pi}) \propto \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1}, \tag{2.11}$$

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\boldsymbol{\mu}|\boldsymbol{\Sigma})p(\boldsymbol{\Sigma}), \tag{2.12}$$

$$= \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_0, \frac{1}{\beta_0}\tilde{\boldsymbol{\Sigma}}_k\right) \mathcal{W}^{-1}(\tilde{\boldsymbol{\Sigma}}_k | \boldsymbol{\Sigma}_0, \nu_0), \tag{2.13}$$

where α_0 is the weight concentration prior, $\boldsymbol{m}_0$ is the mean prior, β_0 is the mean precision prior, $\boldsymbol{\Sigma}_0$ is the covariance prior and ν_0 is the degrees of freedom prior.

Under mild assumptions, it can be shown that the posterior distribution of the parameters is of the same form as the prior [15], thanks to the use of conjugate priors:

$$p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}|\boldsymbol{X}) = q^*(\boldsymbol{\pi})q^*(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{with} \tag{2.14}$$

$$q^*(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_k) \tag{2.15}$$

$$q^*(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_k | \boldsymbol{m}_k, \frac{1}{\beta_k}\tilde{\boldsymbol{\Sigma}}_k\right) \mathcal{W}^{-1}(\tilde{\boldsymbol{\Sigma}}_k | \boldsymbol{\Sigma}_k, \nu_k), \tag{2.16}$$

where α_k, $\boldsymbol{m}_k$, β_k, $\boldsymbol{\Sigma}_k$ and ν_k are obtained using update equations analogous to the EM algorithm for the maximum likelihood solution. We write in Alg.2 the equations of this variational version of the EM algorithm. We refer the reader to Sec.10.2 of [15] for the full derivations of those equations.

Algorithm 2: Variational EM algorithm for Bayesian Gaussian mixtures

Data: a $N \times D$ dimensional dataset of observations $\boldsymbol{X}$, the maximum number of Gaussians K, the prior parameters α_0, $\boldsymbol{m}_0$, β_0, $\boldsymbol{\Sigma}_0$ and ν_0

Result: Posterior parameters α_k, $\boldsymbol{m}_k$, β_k, $\boldsymbol{\Sigma}_k$ and ν_k

1. Initialize three statistics of the data: the number of points N_k, the means $\overline{\boldsymbol{x}}_k$ and the covariances $\boldsymbol{S}_k$ of each cluster (typically with the K-Means algorithm) and deduce the initial α_k, $\boldsymbol{m}_k$, β_k, $\boldsymbol{\Sigma}_k$ and ν_k parameters (see equations in step 3)

2. **Expectation step** Calculate the *responsibilities* r_{nk} using the current parameters values:

$$r_{nk} \propto \tilde{\pi}_k \tilde{\Lambda}_k^{1/2} \exp\left(-\frac{D}{2\beta_k} - \frac{\nu_k}{2}(\boldsymbol{x}_n - \boldsymbol{m}_k)^\top \Sigma_k^{-1}(\boldsymbol{x}_n - \boldsymbol{m}_k)\right), \text{ with}$$

$$\sum_{k=1}^{K} r_{nk} = 1 \text{ for all } n \text{ in } [1, N]$$

$$\log \tilde{\Lambda}_k^{1/2} = \sum_{i=1}^{D} \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D \log 2 + \log |\boldsymbol{\Sigma}_k^{-1}|$$

$$\log \tilde{\pi}_k = \psi(\alpha_k) - \psi\left(\sum_{k=1}^{K} \alpha_k\right), \text{ where } \psi(\cdot)\text{is the digamma function.}$$

3. **Maximization step** Re-evaluate the parameter values using the current *responsibilities*. First recalculate the three statistics of the data:

$$N_k = \sum_{n=1}^{N} r_{nk}$$

$$\overline{\boldsymbol{x}}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n$$

$$\boldsymbol{S}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)^\top.$$

Then recalculate the parameters of the posterior distribution:

$$\alpha_k = \alpha_0 + N_k$$
$$\beta_k = \beta_0 + N_k$$
$$\nu_k = \nu_0 + N_k$$
$$\boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_0 \boldsymbol{m}_0 + N_k \overline{\boldsymbol{x}}_k)$$
$$\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_0 + N_k \boldsymbol{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_0)(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_0)^\top$$

4. Check for convergence of the parameters, if they have not converged return to step 2.

3 Fourier Movement Primitives: an approach for learning robot skills from demonstrations

In this chapter we propose a Fourier movement primitive (FMP) representation to learn robot skills from human demonstrations. We focus here on rhythmic movements for which the method is naturally suited, but we will discuss how it could be applied to discrete (point-to-point) motions. Indeed, we believe that, whether in factory or household scenarios, rhythmic movements play a crucial role in many daily-life tasks. Our approach takes inspiration from the probabilistic movement primitives (ProMP) framework, and is grounded in signal processing theory through the Fourier transform. It works with minimal preprocessing, as it does not require demonstration alignment nor finding the frequency of demonstrated signals. Additionally, it does not entail the careful choice/parameterization of basis functions, that typically occurs in most forms of movement primitive representations. Indeed, its basis functions are the Fourier series, which can approximate any periodic signal. This makes FMP an excellent choice for tasks that involve a superposition of different frequencies. Finally, FMP shows interesting extrapolation capabilities as the system has the property of smoothly returning back to the demonstrations (e.g. the limit cycle) when faced with a new situation, being safe for real-world robotic tasks. We validate FMP in several experimental cases with real-world data from polishing and 8-shape drawing tasks as well as on a 7-DoF, torque-controlled, Panda robot.

3.1 Introduction

Upper-body rhythmic movements play a crucial role in many daily-life tasks. Whether in factory scenarios (e.g. polishing, sawing) or household (e.g. whisking, hammering, wiping), such tasks require the use of repetitive patterns that should adapt to new situations. As opposed to discrete motions (e.g. reaching, picking, batting), where the final location is typically used as the parameter to adapt the task, rhythmic skills contain richer information pertaining to aspects like frequency, amplitude and phase, which can strongly depend on various types of inputs, such as the task context (e.g. wiping a small or large surface). The high number of aspects that need to be accounted for in rhythmic motions make them hard to pre-program. We propose to rely on learning from demonstration (LfD) [13] to learn these rich features.

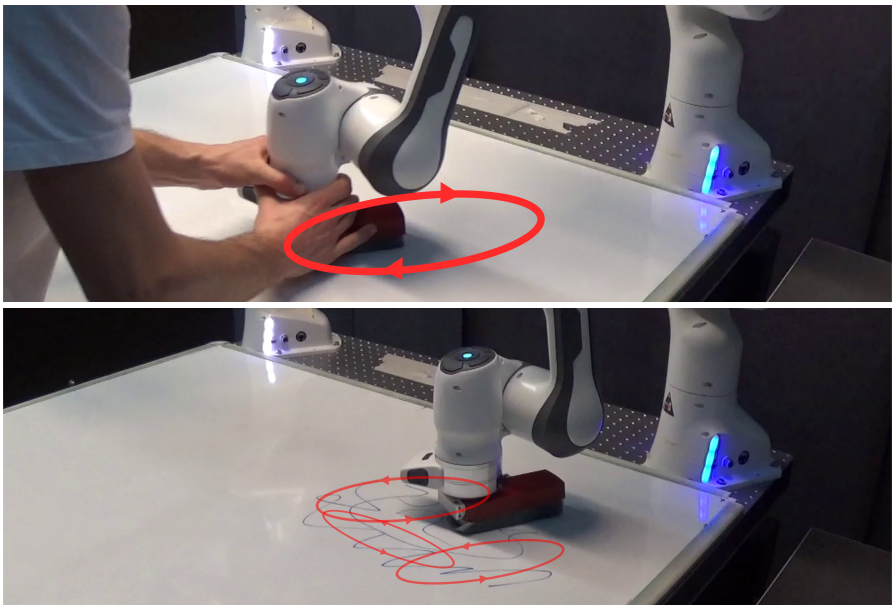


Figure 3.1 – Rythmic tasks such as wiping need to be demonstrated in few demonstrations (**top**), while the robot should extract the important motion features (e.g. amplitude, frequency and phase) and generalize it in a consistent, safe manner (**bottom**).

The problem of learning rhythmic robot skills from demonstrations has received previous attention from the community, especially in the context of wiping/polishing tasks [5, 4, 71, 2], with results along two major research lines. The first one relies on dynamical system representations, through the popular dynamic movement primitives (DMP) [64]. Indeed, extensions of the original DMP [65, 52, 112, 41, 111] have exploited either periodic basis functions or non-linear oscillators to encode demonstrated robot motions. The second, and more recent, line of research leverages probabilistic approaches, either using probabilistic movement primitives (ProMP) [107] or kernelized movement primitives (KMP) [62]. In all cases, sinusoidal basis functions are used, capturing the periodic aspect, but limiting the applicability in cases of varying amplitude, frequency and phase (Section 3.2).

Fourier series have been used extensively during the past decades for synthesis and analysis of periodic signals (Section 3.3). We here propose to leverage them in the context of LfD. The contribution of this chapter is a model for learning rhythmic skills from demonstrations and adapting them to new situations based on Fourier movement primitives (FMP). We propose FMP as a movement primitive representation that relies on a superposition of *Fourier basis functions* (Section 3.4), or complex exponentials, as opposed to the typical choice of real-value sine/cosine basis functions. The main advantages of FMP over the state-of-the-art are:

1. **Extraction of multiple frequencies underlying demonstrations** - by relying on Fourier series as a basis representation, FMP can extract the superposition of various frequencies in a straightforward manner.

2. **No manual choice/tuning of the basis functions** - Fourier basis functions do not require

hyperparameters, in contrast to Von-Mises or sinusoidal basis functions requiring centers, bandwidths and frequencies parameters. The use of the Fourier basis functions is also well motivated theoretically, as any periodic signal can be represented in the Fourier domain.

3. **Minimal preprocessing** - FMP requires very little preprocessing. Namely, it does not require the demonstrations to be aligned, or the basis frequency of the signal to be identified.

4. **Unified magnitude and phase statistics** - the underlying processing with complex numbers allows the system to achive a statistical analysis over **amplitude**, **frequency** and **phase** (illustrated in Fig.3.2).

We evaluate FMP in 3 different scenarios (Section 3.5). First we consider data from a polishing task, requiring one single frequency per degree-of-freedom (DOF). Second we consider the drawing of an 8-shape, which needs a superposition of different frequencies. Finally, we use a 7-DOF Panda robot to perform a whiteboard-wiping task, showing that the robot can start from arbitrary locations in the workspace while smoothly converging to the demonstrations and perform the task. We close the chapter with a discussion on the obtained results (Section 3.6) and conclusion (Section 3.7).

3.2 Related work

In this section we build upon 2.1.1 by reviewing specifically the related work on the representation and learning of periodic movement primitives by imitation, and we place the contribution of this chapter in the context of the state-of-the-art.

3.2.1 Dynamical-system-based approaches

A prominent line of research based on dynamical systems stems from the seminal work of [64] on DMP. The original DMP formulation [64] relies on simple second order dynamics to learn point-attractor movements, while exhibiting interesting properties such as convergence to a desired final state and resistance to perturbations. Owing to a non-linear term that shapes the dynamics, DMP can imitate the shape of demonstrations in a straightforward way. It can be used for both discrete and periodic movements [65], by considering non-linear oscillators and phase dynamics. Following from these results, more complex paradigms in robotics emerged, such as *central pattern generators* [34] and *adaptive frequency phase oscillators* [116].

In [52], Gams et al. exploit the capabilities of adaptive frequency oscillators proposed by [116] in combination with periodic DMP. They propose a two-layered approach that relies firstly on a set of adaptive frequency oscillators to identify the *fundamental* frequency and phase of a demonstrated signal without prior knowledge of its frequency. In a second layer, a periodic DMP is trained using the previously extracted fundamental frequency and phase, to obtain the waveform of the signal, allowing for reproducing the skill with the aforementioned DMP properties. This approach

has been further utilized by others in task generalization [139], human-robot collaboration [111, 110, 113], force control [51] and improved for automatic frequency extraction [112]. These approaches share the limitation that it is not straightforward to perform statistics on the learned model when there is access to multiple demonstrations. This consequently limits the potential of application in compliant control, especially at the level of *minimal intervention control* [90, 19, 137]. Compliant control is possible using such kind of dynamical systems, however the control policies do not reflect the structure of the data and are typically modulated by external signals, such as EMG [111].

Finally, [2, 72, 71] propose to use *autonomous* dynamical systems to learn polishing tasks, relying on formulations that share similarities with [70]. In these works, learning is done to the extent that the robot extracts surface normals [2] and adapts its behavior to new human intentions (either through different limit cycles [72] or task switches [71]). We, instead, focus on the learning of the spatiotemporal aspects of demonstrations, namely magnitude, frequency and phase.

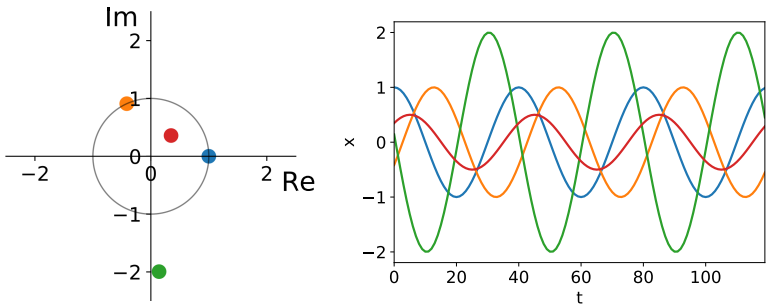
3.2.2 Probabilistic approaches

While probabilistic approaches for motor primitive learning by imitation rose in popularity, two lines of approaches gain particular relevance for rhythmic skills. The probabilistic movement primitives [107] presented in previous chapter can represent either discrete or periodic motions, depending on the choice of the basis functions. By relying on cosine or Von-Mises basis functions, ProMP can represent periodic motions [105], but has limited adaptation capability in terms of frequency and phase.

In another direction, following the spirit of non-parametric learning, Gaussian process regression (GPR) can also model periodic time series (see [114] ch. 4), and hence can also approximate well rhythmic robot skills, by relying on appropriate kernels. However, it is computationally expensive and it is not straightforward to adapt a demonstrated policy to a new situation. More recently, kernelized movement primitives (KMP) [62, 61] have been shown to permit the learning of periodic skills when using periodic kernel functions. Nonetheless, both KMP and GPR, despite allowing for statistics, share the same limitations as ProMP in that the kernels conventionally employed are not expressive enough to represent a wide range of frequencies and phases.

3.2.3 Constraint-based approaches

A third relevant line of research focuses on learning motion constraints [5, 4, 84] through the estimation of null space matrices from data. While [5, 84] perform polishing/wiping on flat surfaces, [4] extend the approach to be compatible with curved surfaces (which is also the motivation behind [2]). Similarly to [2, 72, 71], the focus is not on the learning of rhythmic motion primitives, hence application to tasks involving periodic motions (e.g. drumming, hammering) is not straightforward. However, these approaches rely on policy learning for generalizing the learned constraints. Hence, there is a high potential for combinations with FMP in the future.



(a) Weights in the complex plane (b) Corresponding basis functions multiplied by weights

Figure 3.2 – Illustration of reconstructed signals with one Fourier basis function (for $k = 3$). The orange and blue points have the same amplitude but not the same phase, which results in the same signals that are shifted in time.

3.3 Preliminaries

We briefly recall the concepts of discrete Fourier transform and inverse discrete Fourier transform, which are used to convert sequences from time domain to frequency domain, and the other way around.

3.3.1 Discrete Fourier transform

The discrete Fourier transform converts a one-dimensional sequence $\boldsymbol{y} = [y_0, \ldots, y_{T-1}]^\top$ of T equally-spaced samples into a same length sequence of complex coefficients corresponding to different frequencies. The basic idea is to consider the sequence $\boldsymbol{y}$ as a periodic signal of period T [I]. The sequence can be perfectly represented in the frequency domain with T complex coefficients:

$$\forall k \in [\![0; T-1]\!] : \tilde{w}_k = \sum_{n=0}^{T-1} y_n \exp\left(-\frac{2i\pi}{T} kn\right), \tag{3.1}$$

where i refers to the imaginary part of a complex number. By concatenating the T coefficients in a vector, we get the following matrix-form formula:

$$\begin{gathered} \tilde{\boldsymbol{w}} = \boldsymbol{\Psi}\boldsymbol{y} \quad \text{with:} \\ \forall (k,n) \in [\![0;T-1]\!]^2 : \Psi_{k,n} = \exp\left(-\frac{2i\pi}{T} kn\right). \end{gathered} \tag{3.2}$$

[I] For discrete movements, a periodic signal of period 2T can be constructed by symmetrizing the original signal of length T, so that the same method can be applied.

3.3.2 Inverse discrete Fourier transform

The discrete Fourier transform is an invertible, linear transformation. Therefore, we can map the frequency domain representation of the signal back to the time domain:

$$\forall n \in [\![0; T-1]\!] : y_n = \frac{1}{T} \sum_{k=0}^{T-1} \tilde{w}_k \exp\left(\frac{2i\pi}{T} kn\right). \tag{3.3}$$

This can also be expressed in matrix form as:

$$\boldsymbol{y} = \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{w}} \quad \text{with} \quad \tilde{\boldsymbol{\Phi}} = \frac{1}{T} \boldsymbol{\Psi}^H, \tag{3.4}$$

where H denotes the Hermitian transpose operator, which is obtained by taking the tranpose and then taking the complex conjugate of each entry:

$$(\boldsymbol{\Psi}^H)_{ij} = \overline{\boldsymbol{\Psi}_{ji}}, \tag{3.5}$$

with the overbar denoting the scalar complex conjugate. An interesting property of Fourier basis functions is that a single basis function represents variations of amplitude and phase. We illustrated this in Fig. 3.2, where we can indeed see that the same basis function can represent signals of different amplitudes (see red, blue and green signals), but also the same signal shifted in time (see blue and orange signals).

3.4 Fourier movement primitives

In this section, we present Fourier movement primitives. First, we detail how we can compute statistics from demonstrations, then we explain how this is exploited for minimal intervention control in the Fourier domain.

3.4.1 Imitation learning

Let $(\boldsymbol{y}_l)_{l=1,\dots,N}$ be a series of N demonstrations of length T. For clarity purposes, we assume that the demonstrations contain only one degree of freedom (we will discuss in subsection (3.4.4) how it is extended to multiple ones). We compute using (3.4) the complex weights $(\tilde{\boldsymbol{w}}_l)_{l=1,\dots,N}$ such that

$$\forall l \in [\![1; N]\!] : \boldsymbol{y}_l = \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{w}}_l. \tag{3.6}$$

We then learn a distribution of $(\tilde{\boldsymbol{w}}_l)_{l=1,\dots,N}$. The main difference here, with respect to standard ProMP, is that the weights are complex numbers. As we want to have correlations between real and imaginary parts of our weights (so that we can learn correlations in magnitudes or phases), we consider an expanded real version of our weights where the real and imaginary parts are

concatenated as:

$$\boldsymbol{w}_l = [\mathrm{Re}(\tilde{\boldsymbol{w}}_l)^\top, \mathrm{Im}(\tilde{\boldsymbol{w}}_l)^\top]\,. \tag{3.7}$$

It is straightforward to see that $\boldsymbol{w}_l$ and $\tilde{\boldsymbol{w}}_l$ are linear in the complex space:

$$\tilde{\boldsymbol{w}}_l = \boldsymbol{A}\boldsymbol{w}_l \text{ with } \boldsymbol{A}_{T\times 2T} = \begin{bmatrix} \boldsymbol{I}_T & i\boldsymbol{I}_T \end{bmatrix}. \tag{3.8}$$

For notation simplicity, we define $\boldsymbol{\Phi} = \boldsymbol{A}\tilde{\boldsymbol{\Phi}}$, which implies:

$$\forall l \in [\![1; N]\!] : \boldsymbol{y}_l = \boldsymbol{\Phi}\boldsymbol{w}_l. \tag{3.9}$$

We learn the distribution of the weights $(\boldsymbol{w}_l)_{l=1\,\ldots\,N}$ by fitting a Gaussian mixture using the Expectation-Maximization algorithm, initialized with the K-means algorithm. We retrieve the weights, means and covariances $\boldsymbol{\theta} = (\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)_{j=1,\ldots,M}$ of the Gaussian mixture, whose probability density function is expressed as:

$$\begin{aligned} p(\boldsymbol{w}|\theta) &= \sum_{j=1}^{M} \pi_j \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \\ \text{with } \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) &= \frac{1}{(2\pi)^{(2T)/2}|\boldsymbol{\Sigma}_j|^{1/2}} \exp\big\{-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_j)\big\}. \end{aligned} \tag{3.10}$$

We would like to give here an illustration of the approach on toy data, to highlight the usefulness of using Fourier basis functions compared to other types of basis functions, and to give an intuition of why demonstration alignment is not required when using such basis functions. We show in Fig.3.3a twenty toy demonstrations for a point-to-point movement. The variability in the demonstrations is illustrative of the type of variations one has when showing point-to-point movements, and of why a demonstration alignment is usually necessary. In Fig.3.3e, the demonstrations are symmetrized to make them periodic, and a mixture of 5 Gaussians is learned in the Fourier domain on those periodic demonstrations. Samples from the distribution learned with our approach are shown in Fig.3.3f. A typical choice of demonstrations for representing point-to-point movements is Radial Basis Functions, we show in Fig.3.3b the set of 20 radial basis functions that we chose. Demonstrations are mapped to the lower-dimensional subspace of dimension 20, and a mixture of 5 Gaussians is learned in this latent space. We show in Fig.3.3d samples from the distribution learned. Notably, we can see that radial basis functions result in a very poor representation of the demonstrations, whereas Fourier basis functions can approximate the distribution very well. Intuitively, this is because Fourier basis functions can permit to make statistics over both magnitude and phase, and hence can deal with non-aligned demonstrations. It is worth verifying that the set of radial basis functions chosen is good enough for representing the demonstrations, we plot in Fig.3.3c the reconstruction of the demonstrations mapped through the radial basis functions forth and back. We can see that the demonstrations have indeed been encoded properly, so the bad distribution learned with the radial basis functions is not due to a bad choice of the radial basis functions. We highlight here again that the appropriateness of the basis functions does not have to be checked with Fourier basis functions because of the

equivalence between time and frequency domain which guarantees that the demonstrations are perfectly represented in the Fourier domain.

We will use the distribution learned in the Fourier domain to perform minimal intervention control [138]. To do so, we need a way to transform a partial trajectory (e.g., the starting position of the robot) to the Fourier domain. In the context of ProMP, this is typically done by conditioning on the distribution (usually a single Gaussian). We observed that this is not suitable for the high number of dimensions we have, hence we propose a different approach that scales better with the number of dimensions.

3.4.2 Mapping partial trajectories to Fourier domain

Given a partial demonstration $\boldsymbol{y}_{1:K}$ of size K, we search $\boldsymbol{w}$ such that:

$$\boldsymbol{y}_{1:K} = \boldsymbol{\Phi}_{1:K}\boldsymbol{w}, \tag{3.11}$$

with $\boldsymbol{\Phi}_{1:K}$ of size $K \times T$ (containing the first K rows of $\boldsymbol{\Phi}$). It is important to note that the approach is also valid for partial trajectories that do not occur at the beginning of the movement, or arbitrary keypoints.

A straightforward, but naive, solution would be to choose $\boldsymbol{w} = (\boldsymbol{\Phi}_{1:K})^{+}\boldsymbol{y}_{1:K}$, which, in practice, results in a value for $\boldsymbol{w}$ that is far from the distribution of demonstrated data, resulting in poor tracking. We instead propose to leverage the knowledge of the demonstrations distribution in the Fourier domain (as learned in Section 3.4.1) to find a set of weights $\boldsymbol{w}$ that is close to the demonstrations, while respecting (3.11).

This can be written as the optimization problem:

$$\max_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{\theta}) \quad \text{s.t.} \quad \boldsymbol{y}_{1:K} = \boldsymbol{\Phi}_{1:K}\boldsymbol{w}, \tag{3.12}$$

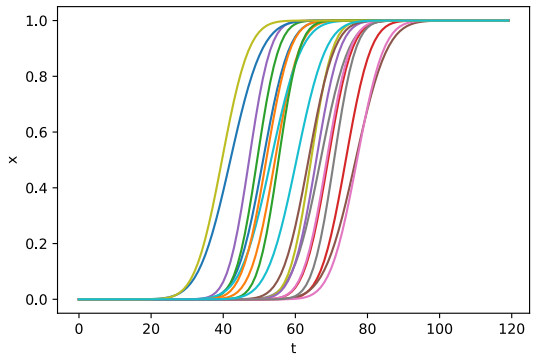
which is equivalent to:

$$\min_{\boldsymbol{w}} \big(-\log p(\boldsymbol{w}|\boldsymbol{\theta})\big) \quad \text{s.t.} \quad \boldsymbol{y}_{1:K} = \boldsymbol{\Phi}_{1:K}\boldsymbol{w}. \tag{3.13}$$

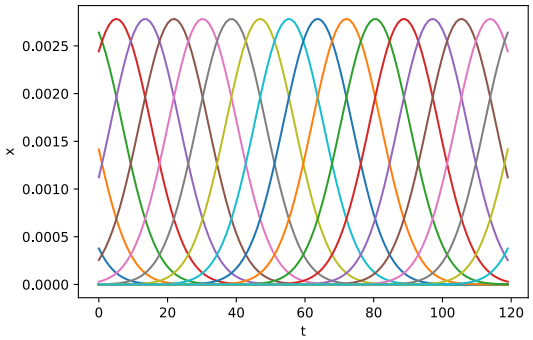
To solve this problem more efficiently, we use a Lagrangian relaxation:

$$\min_{\boldsymbol{w}} \big(\|\boldsymbol{y}_{1:K} - \boldsymbol{\Phi}_{1:K}\boldsymbol{w}\|^2 - \lambda \log p(\boldsymbol{w}|\boldsymbol{\theta})\big), \tag{3.14}$$

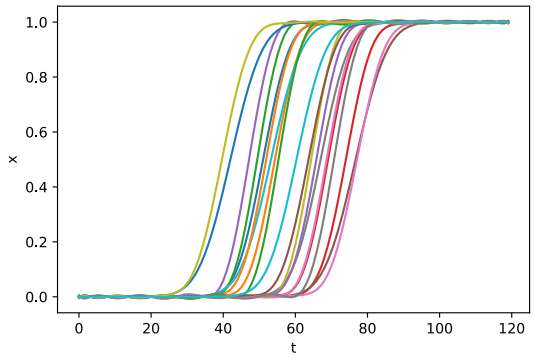
where λ is the Lagrange multiplier. We could find the value of λ by solving the Lagrangian dual problem, but for simplicity purposes we fix λ to an arbitrary small value ($1e-8$) as it yields good results in all of our experiments. In practice, the weights $\boldsymbol{w}$ are of high dimensions and therefore the different Gaussians of the mixture have no overlap (formally, this means that the mutual information between any two Gaussians of the mixture is almost zero). The solution of (3.14) must verify that $p(\boldsymbol{w}|\boldsymbol{\theta})$ is not numerically zero (otherwise $-\lambda \log p(\boldsymbol{w}|\boldsymbol{\theta})$ tends to



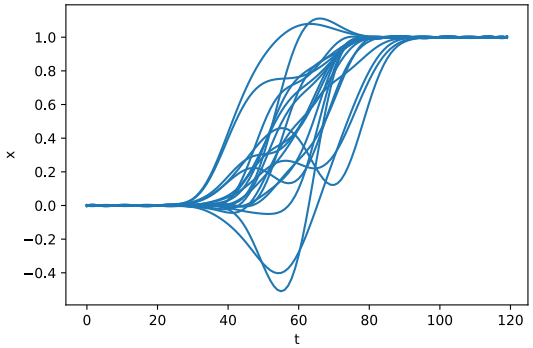
(a) Toy Demonstrations



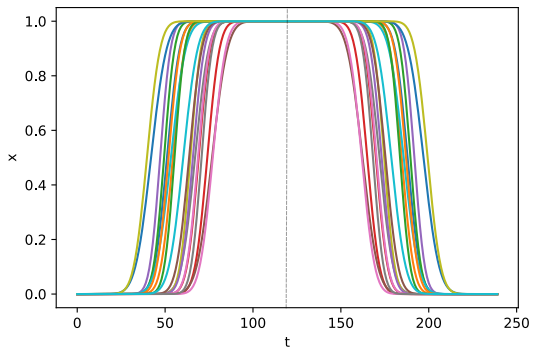
(b) Radial basis functions



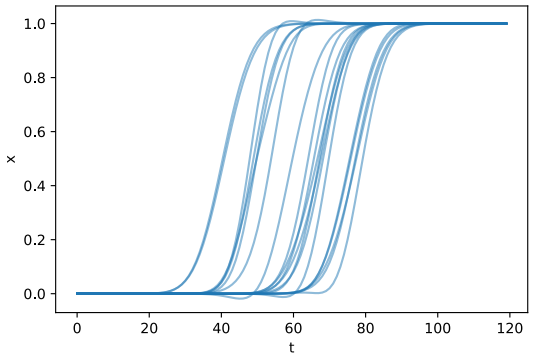
(c) Reconstruction of the demonstrations with radial basis functions



(d) Samples from distribution learned in ProMP latent space (with a mixture of 5 Gaussians)



(e) Demonstrations symmetrized (the first 120 timesteps contain the demonstrations and the last 120 timesteps the respective time-reversed demonstrations)



(f) Samples from distribution learned in the Fourier domain (with a mixture of 5 Gaussians), truncated over the first half of the time domain

Figure 3.3 – Illustration of the advantage of Fourier basis functions for non-aligned point-to-point demonstrations

infinity). Under the hypothesis that the Gaussians have almost-zero mutual information, we can find candidate solutions by solving M least squares problems:

$$\begin{aligned}\boldsymbol{w}^j &= \arg\min_{\boldsymbol{w}} \left(\|\boldsymbol{y}_{1:K} - \boldsymbol{\Phi}_{1:K}\boldsymbol{w}\|^2 - \lambda \log \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right) \\ &= \arg\min_{\boldsymbol{w}} \left(\|\boldsymbol{y}_{1:K} - \boldsymbol{\Phi}_{1:K}\boldsymbol{w}\|^2 + \lambda \|\boldsymbol{w} - \boldsymbol{\mu}_j\|^2_{\boldsymbol{\Sigma}_j^{-1}} \right) \\ &= \left(\boldsymbol{\Phi}_{1:K}^H \boldsymbol{\Phi}_{1:K} + \lambda \boldsymbol{\Sigma}_j^{-1} \right)^{-1} \left(\boldsymbol{\Phi}_{1:K}^H \boldsymbol{y}_{1:K} + \lambda \boldsymbol{\Sigma}_j^{-1} \boldsymbol{\mu}_j \right).\end{aligned} \tag{3.15}$$

We can then solve (3.14) by finding the minimum over the finite set of solutions $(\boldsymbol{w}^j)_{j=1}^M$:

$$j^* = \arg\min_{j \in [\![1;M]\!]} \left(\|\boldsymbol{y}_{1:K} - \boldsymbol{\Phi}_{1:K}\boldsymbol{w}^j\|^2 - \lambda \log(\pi_j) + \lambda \|\boldsymbol{w} - \boldsymbol{\mu}_j\|^2_{\boldsymbol{\Sigma}_j^{-1}} \right), \tag{3.16}$$

which allows us to map our partial trajectory to the Fourier domain with:

$$\boldsymbol{w}_K = \boldsymbol{w}^{j^*}. \tag{3.17}$$

The full process is summarized in Algorithm 3. Next, we propose a tracking controller in the

Algorithm 3: Partial trajectory mapping

Data: Partial observations $\boldsymbol{y}_{1:K}$ up to timestep K
Result: Fourier weight $\boldsymbol{w}_K$ such that $\boldsymbol{y}_{1:K} \simeq \boldsymbol{\Phi}_{1:K}\boldsymbol{w}_K$
Find M candidate solutions $(\boldsymbol{w}^j)_{j=1}^M$ with Eq. (3.15)
Compute minimum $\boldsymbol{w}_K$ with Eqs. (3.16)-(3.17)

Fourier domain, leveraging the distribution learned and the possibility to map partial trajectories to the Fourier domain.

3.4.3 Tracking in the Fourier domain

The ability to do minimal intervention control in the Fourier domain is a core component of our proposed method, as it permits to modulate both phase and amplitude by exploiting the variability of the provided demonstrations. We will track only one Gaussian for simplicity purposes (the solution of Eq.3.16). This seems to be a reasonable assumption because in high dimensions, the different Gaussians in the mixture are likely to have a very small overlap. We track this Gaussian in the Fourier domain with the given covariance.

We could do this by using model predictive control (MPC) in the Fourier domain, but, as the number of dimensions is high (T is the trajectory length), it would be too computationally expensive. We propose to use a simple proportional controller to track in the Fourier domain: an approach that proves satisfactory in practice. Given a current trajectory up to timestep t, represented as $\boldsymbol{w}_t$ in the Fourier space, we track the target $\boldsymbol{\mu}_{j^*}$ with precision matrix $\boldsymbol{\Sigma}_{j^*}^{-1}$. The

update rule of the tracking controller is:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \text{dt}\ \beta\ \text{diag}(\boldsymbol{\Sigma}_{j^*}^{-1})(\boldsymbol{\mu}_{j^*} - \boldsymbol{w}_t), \tag{3.18}$$

where $\text{diag}(\cdot)$ is an operator zeroing all offdiagonal elements. We choose to weigh the updates by $\text{diag}(\boldsymbol{\Sigma}_{j^*}^{-1})$ and not $\boldsymbol{\Sigma}_{j^*}^{-1}$ because we observed potential instabilities with the latter in practice. More sophisticated controllers could be used to leverage the full-rank structure of the precision matrix $\boldsymbol{\Sigma}_{j^*}^{-1}$, and we shall address this in future work.

Similarly to ProMP, we can go back from Fourier domain to time domain and find the next point to track as well as the appropriate tracking covariance:

$$\begin{aligned} \boldsymbol{y}_{t+1}^{\text{des}} &= \boldsymbol{\Phi}_{t+1}\boldsymbol{w}_{t+1}, \\ \boldsymbol{\Sigma}_{\boldsymbol{y}_{t+1}^{\text{des}}} &= \boldsymbol{\Phi}_{t+1}\boldsymbol{\Sigma}_{j^*}\boldsymbol{\Phi}_{t+1}^H, \end{aligned} \tag{3.19}$$

with $\boldsymbol{\Phi}_{t+1}$ of size $1 \times T$ (containing the $(t+1)^{\text{th}}$ row of $\boldsymbol{\Phi}$).

The pseudocode of the algorithm is given in Algorithm 4.

Algorithm 4: Tracking in Fourier domain

Data: Partial observations $\boldsymbol{y}_{1:K}$ up to timestep K
Result: desired trajectory $\boldsymbol{y}_{K+1:T}^{\text{des}}$ for timesteps $K+1$ to T and desired covariances $[\boldsymbol{\Sigma}_{\boldsymbol{y}_{K+1}^{\text{des}}}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{y}_{T}^{\text{des}}}]$
Calculate $\boldsymbol{w}_K$ using Eqs.(3.15)-(3.17)
for $t \leftarrow K$ **to** $T-1$ **do**
  Calculate $\boldsymbol{w}_{t+1}$ using Eq.(3.18)
  Calculate $\boldsymbol{y}_{t+1}^{\text{des}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{y}_{t+1}^{\text{des}}}$ using Eq.(3.19)
end

3.4.4 Multidimensional case

We discuss here the extension of our method to several degrees of freedom D. The extension is straightforward as it consists of concatenating along the dimensions. Following the previous notation, the data and partial data are written as such:

$$\boldsymbol{y}_i = \begin{pmatrix} \boldsymbol{y}_i^1 \\ \vdots \\ \boldsymbol{y}_i^D \end{pmatrix} \quad \text{and} \quad \boldsymbol{y}_{1:K} = \begin{pmatrix} \boldsymbol{y}_{1:K}^1 \\ \vdots \\ \boldsymbol{y}_{1:K}^D \end{pmatrix}, \tag{3.20}$$

where the superscript j of $\boldsymbol{y}_i^j$ denotes the j^{th} degree of freedom. And the $\boldsymbol{\Phi}$ matrix is used to construct a block-diagonal matrix with D entries:

$$\boldsymbol{\Phi}^D = \begin{pmatrix} \boldsymbol{\Phi} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \boldsymbol{\Phi} \end{pmatrix}. \tag{3.21}$$

Similarly, $\boldsymbol{\Phi}_{1:K}$ and $\boldsymbol{\Phi}_t$ are concatenated D times block-diagonally. It is worth noting that in this case, $\boldsymbol{w}$ is a vector of length TD, which means that the Gaussian mixture learned captures correlations between the different degrees of freedom.

3.5 Experiments

In this section we show the performance of FMP on various datasets. First, we describe the data acquisition and preprocessing step. Then, a polishing task and the task of drawing a 8-shape are presented. Finally, the task of wiping a whiteboard is considered and applied on a real robot. When applicable, our method will be compared against the use of ProMP with Von-Mises basis functions. Videos of the experimental evaluation can be found at https://sites.google.com/view/fourier-movement-primitives.

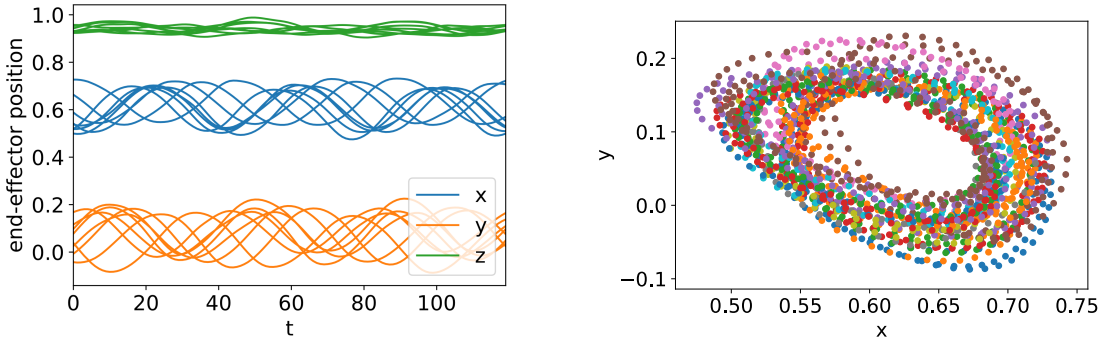
3.5.1 Data acquisition and preprocessing

For simplicity and visualization purposes, in all tasks the data consists of the position of the robot end-effector, and is therefore 3-dimensional. All demonstrations are obtained by kinesthetically teaching the robot. As the tasks are rhythmic, we propose to reduce the human burden by showing only one (long) demonstration, that is then preprocessed. The demonstration is acquired at 20Hz, and we cut it in subdemonstrations of length T, arbitrarily chosen to 120 in our experiments (corresponding to 6 seconds). To cut the demonstration, we let a sliding window slide across the demonstration by increments of 10 timesteps. By doing so, we exploit the fact that the task is rhythmic and can start anywhere.

3.5.2 Polishing task

The polishing task is a representative example because it can contain as low as one frequency for each degree of freedom. A 3-minute demonstration is recorded with the robot, from which the demonstrations are cut as explained above.

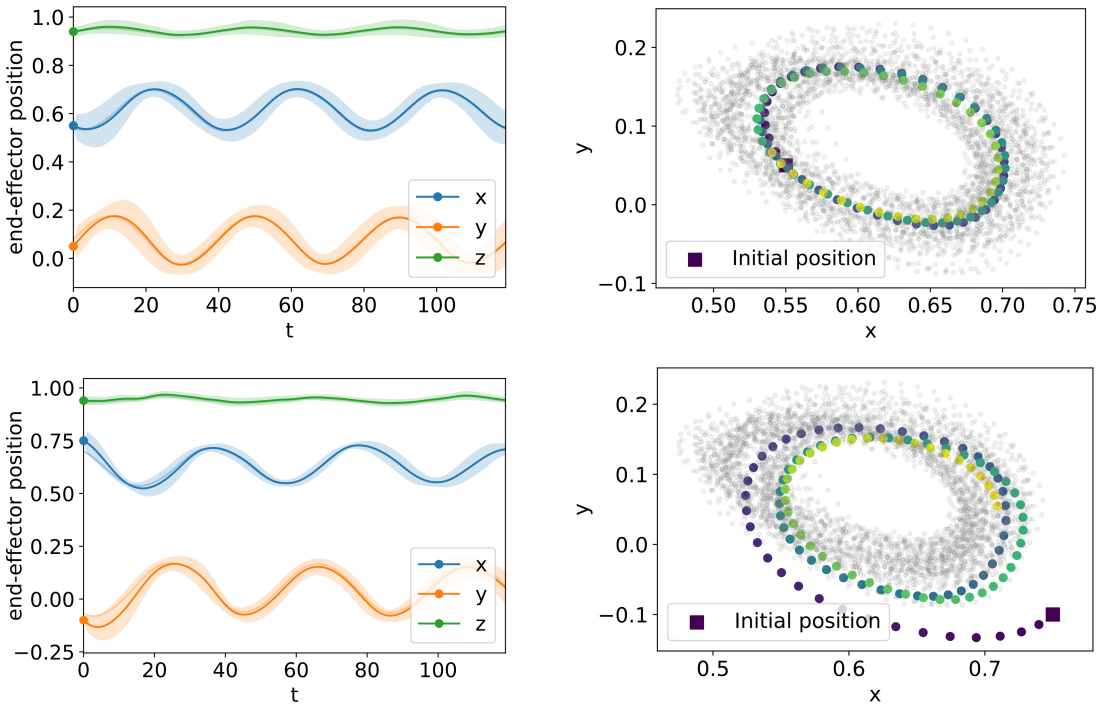
The demonstrations are shown in Fig.3.4. We learn the distribution of the data in the Fourier domain with $M = 10$ Gaussians. We show in Fig.3.5 the tracking for different starting positions: a position that belongs to the data distribution (interpolation), and a position outside of the data distribution (extrapolation).



(a) End-effector position as a function of time

(b) Demonstrations in the $x - y$ plane (different colors represent different demonstrations).

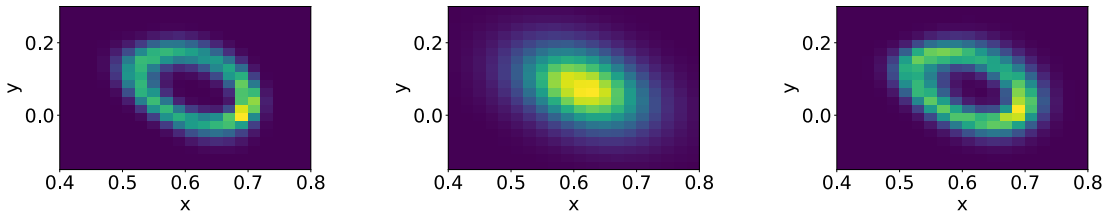
Figure 3.4 – Demonstrations of the polishing task.



(a) Tracking trajectory x, y, z in function of time, along with uncertainties

(b) Scatter plot of the tracking trajectory on dimensions x and y (color gradient for time)

Figure 3.5 – Polishing from different initial positions with FMP.



(a) $T = 40$, alignment, $M = 1$

(b) $T = 120$, no alignment, $M = 1$

(c) $T = 120$, no alignment, $M = 10$

Figure 3.6 – Heatmaps of distribution learned with ProMP-VM.

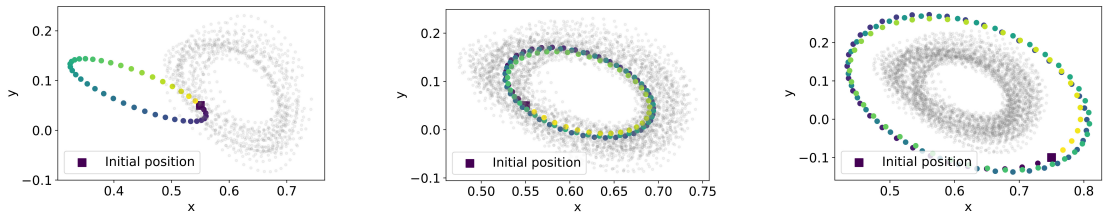
As we can see, our method permits both interpolation and extrapolation with respect to the starting position. We compare now our method to the standard ProMP with Von-Mises basis functions (later abbreviated ProMP-VM) [105]:

$$\begin{aligned} b_i^{\text{VM}}(z_t) &= \exp\left(\frac{\cos\left(2\pi f(z_t - c_i)\right)}{h}\right), \\ \Phi_i(z_t) &= \frac{b_i^{\text{VM}}(z_t)}{\sum_{j=1}^{n} b_j^{\text{VM}}(z_t)}, \end{aligned} \tag{3.22}$$

where f denotes the frequency of the signal, c_i the center of the basis function and h the width. This method requires the demonstrations to be aligned, and to contain exactly one period of the signal. For illustration purposes, we show how ProMP-VM performs after alignment of the data and cutting to contain only one period (roughly at $T = 40$), and therefore $f = 1$. We used 20 basis functions with the centers c_i uniformly placed between 0 and 2π. The hyperparameter h is selected so that the basis functions become cosine (high value of h, as the exponential function is locally equal to the identity around 0, up to a constant). We show in Fig.3.6a a heatmap of the learned distribution, where we can see that, in the case of careful data alignment, ProMP-VM can approximate the distribution of polishing demonstrations well. In the original ProMP method, only one Gaussian is used to approximate the distribution of the demonstrations.

Even if, to the best of our knowledge, this has not been proposed in the ProMP literature, we will show that increasing the number of Gaussians can alleviate the need for demonstration alignment. Indeed, due to higher variability in the phase domain, the distribution of weights becomes multi-modal and hence is more accurately encoded by a mixture. In Fig.3.6b, we show the obtained results for ProMP-VM using more than one period (T=120), where we had to explicitly provide the frequency of the signal (in this case, f=3). As we can see, without alignment, ProMP-VM fails to approximate the distribution of the data. For a fairer comparison, we also extend ProMP-VM by learning the distribution with a mixture model (10 Gaussians) and show in Fig.3.6c that doing so permits to approximate the distribution well.

The results in Fig.3.6 show that, by bringing ProMP closer to FMP, the original ProMP formulation can be greatly improved. However, a major difference between ProMP-VM and FMP lies in the way we generate trajectory distributions that go through keypoints (see 3.4.2). With ProMP-VM, it is done via conditioning, whereas in FMP is is achieved by mapping the keypoint to the Fourier



(a) $T = 40$, alignment, $M = 1$

(b) $T = 120$, no alignment, $M = 10$

(c) $T = 120$, no alignment, $M = 10$

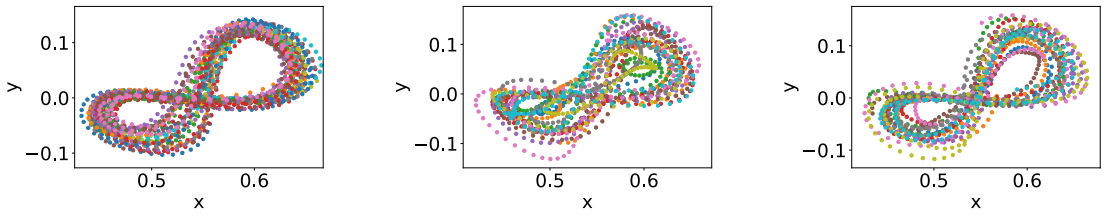
Figure 3.7 – Conditioning on initial position with ProMP-VM.

domain, and tracking in the Fourier domain (here, the keypoint that we evaluate is the starting point, but it is applicable to any keypoint or partial trajectory). Indeed, as shown in Fig.3.7a, when using ProMP-VM with alignment it is not possible to start the movement from a different region than the one observed in the demonstrations. When we use multiple Gaussians without alignment, this adaptation capability becomes possible (Fig.3.7b). Since ProMP-VM represents the demonstrations with periodic basis functions, it can only generate periodic signals that will pass through the initial point. As seen in Fig.3.7c, this mechanism does not allow to cope well with perturbations that require to extrapolate outside of the demonstrations, while following the demonstrations in the next cycles. Indeed, when conditioning outside of the training data, it tends to produce overconfident trajectory distributions (because the Gaussian mixture is learned by maximizing the log-likelihood) that do not return to the demonstrations. In contrast, FMP can generate trajectory distributions that return back to the training data in a way that is compatible with the variations that were observed in the demonstrations, as we can see in Fig.3.5.

3.5.3 8-shape drawing

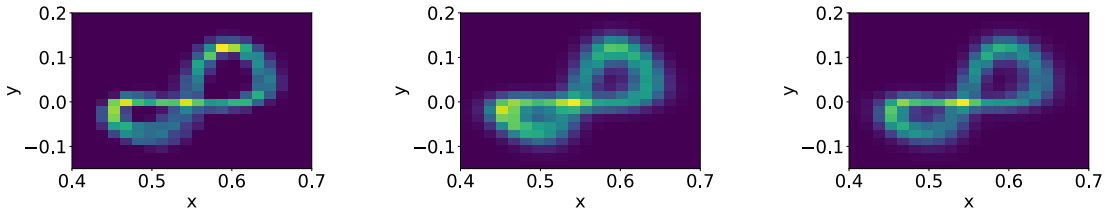
We demonstrate a 3-minute drawing of an 8-shape, used as a standard benchmark task [52, 41]. This task is interesting because it involves a superposition of different frequencies. In the standard ProMP-VM, only one frequency can be approximated. For a better analysis of the performances of FMP, we propose here to benchmark FMP against an extension of ProMP-VM that can approximate a superposition of frequencies. We include basis functions for different frequencies f, namely for f from 1 to 5. For each f, 20 offset basis functions are used, as previously. We use this extension of ProMP-VM on the same data as FMP (no alignment), and with $M = 10$ for a fair comparison. We denote this extension as ProMP-VM-Mult.

Fig.3.8 shows demonstration samples, ProMP-VM-Mult samples, and FMP samples. We observe that FMP samples are smoother and closer to the demonstrations than ProMP-VM-Mult samples, which suggests that the distribution has been better learned with FMP than ProMP-VM-Mult. To verify this observation, we computed a heatmap of the learned distribution. For ProMP-VM-Mult and FMP, we sample 10000 trajectories from the learned distribution, and compute the heatmap. Those are shown in Fig.3.9, next to the demonstrations heatmap.



(a) Demonstrations (b) ProMP-VM-Mult (c) FMP

Figure 3.8 – Samples of the learned 8-shape distribution.



(a) Demonstrations (b) ProMP-VM-Mult (c) FMP

Figure 3.9 – Heatmaps of the learned 8-shape distribution.

We can see that the heatmap of ProMP-VM-Mult is more blurred compared to the FMP heatmap (more samples seem to fall inside the 8 holes). To confirm this, we propose to evaluate quantitatively the learned distribution. We compare the distributions learned with ProMP-VM-Mult and FMP to the ground truth (obtained from the demonstrations). We note Q the ground truth distribution, and P the approximate distribution (respectively obtained with ProMP-VM-Mult or FMP). The distributions are discrete probability distributions, defined over the finite set of cases $\mathcal{X}$ of the heatmap. We considered two different metrics:

- the Forward Kullblack-Leibler divergence:

$$D_{\mathrm{KL}}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}.$$

  Forward KL is known as *zero avoiding*, as it penalizes $Q(x) = 0$ when $P(x) > 0$. This therefore quantifies if the distribution learned covers well the ground truth distribution.

- the Reverse Kullblack-Leibler divergence:

$$D_{\mathrm{KL}}(Q||P) = \sum_{x \in \mathcal{X}} Q(x) \log \frac{Q(x)}{P(x)}.$$

  Reverse KL is known as *zero forcing*, as it does not penalize $Q(x) = 0$ when $P(x) > 0$. This therefore measures how well our distribution Q approximates a part of the ground truth distribution.

	ProMP-VM-Mult	FMP
Forward KL	0.20	**0.11**
Reverse KL	0.53	**0.28**

Table 3.1 – Quantitative comparison of distributions learned with ProMP-VM-Mult and FMP versus ground truth distribution for 8-shape task.

The results are presented in Table 3.1. We observe that FMP has learned a distribution that is about twice closer to the ground truth distribution compared to ProMP-VM-Mult. This can be interpreted easily, as ProMP-VM-Mult has several basis functions for a given frequency, which gives many more basis functions for the same given number of frequencies, resulting in poor statistics. Finally, we evaluate how FMP can generate trajectories that start at any given position. In Fig.3.10, we can see that, even for tasks that involve a superposition of different frequencies, FMP can generate trajectory distributions that get back to the training data in a way that is compatible with the variations observed in the demonstrations. The results with ProMP-VM-Mult were unsatifactory, consistently with Fig.3.8, they have therefore not been included in the manuscript.

3.5.4 Real-world wiping task

Finally, we apply FMP to a real-world robotic task of whiteboard wiping. Our robot is a 7-DoF torque-controlled Panda robot. We record a 2-minute demonstration of whiteboard wiping with kinesthetic teaching. The demonstration is then split into subdemonstrations of length $T = 120$ (6s) as explained previously. For simplicity purposes, only the position of the robot end-effector is recorded, the statistics are therefore made on end-effector position trajectories (with $M = 10$ Gaussians). The robot is then controlled with an impedance controller that tracks the desired trajectory with manually specified gains, with a fixed orientation (we allow the robot to be compliant around the normal to the plane by setting low orientation gains around that axis). An overview of the setup is shown in Fig.3.1. In this experiment we show that we can generate movements of arbitrary durations with FMP. While this should be trivial because we have periodic basis functions over the duration T, this is not in practice as we did not preprocess the data so that the beginning and end of the demonstrations are equal. We alleviate this by recomputing at timestep T the Fourier weights $\boldsymbol{w}$ given the partial trajectory from $T - K$ to T (in practice, we use K=10), and subsequently can use the desired trajectory between timesteps T and $2T - K$. We then repeat this process (it is interesting to note that every time we recompute $\boldsymbol{w}$ using the partial trajectory mapping, we allow the trajectory to change the Gaussian that is tracked). While this might appear cumbersome, this is in practice very efficient, and much easier than having to align the demonstrations. To evaluate the quality of the learned distribution, we propose to show two movements given a desired initial position:

- One where we track the Gaussian mean as explained in Section 3.4.3.
- One where we sample from the Gaussian distribution and track this sample instead of the mean.

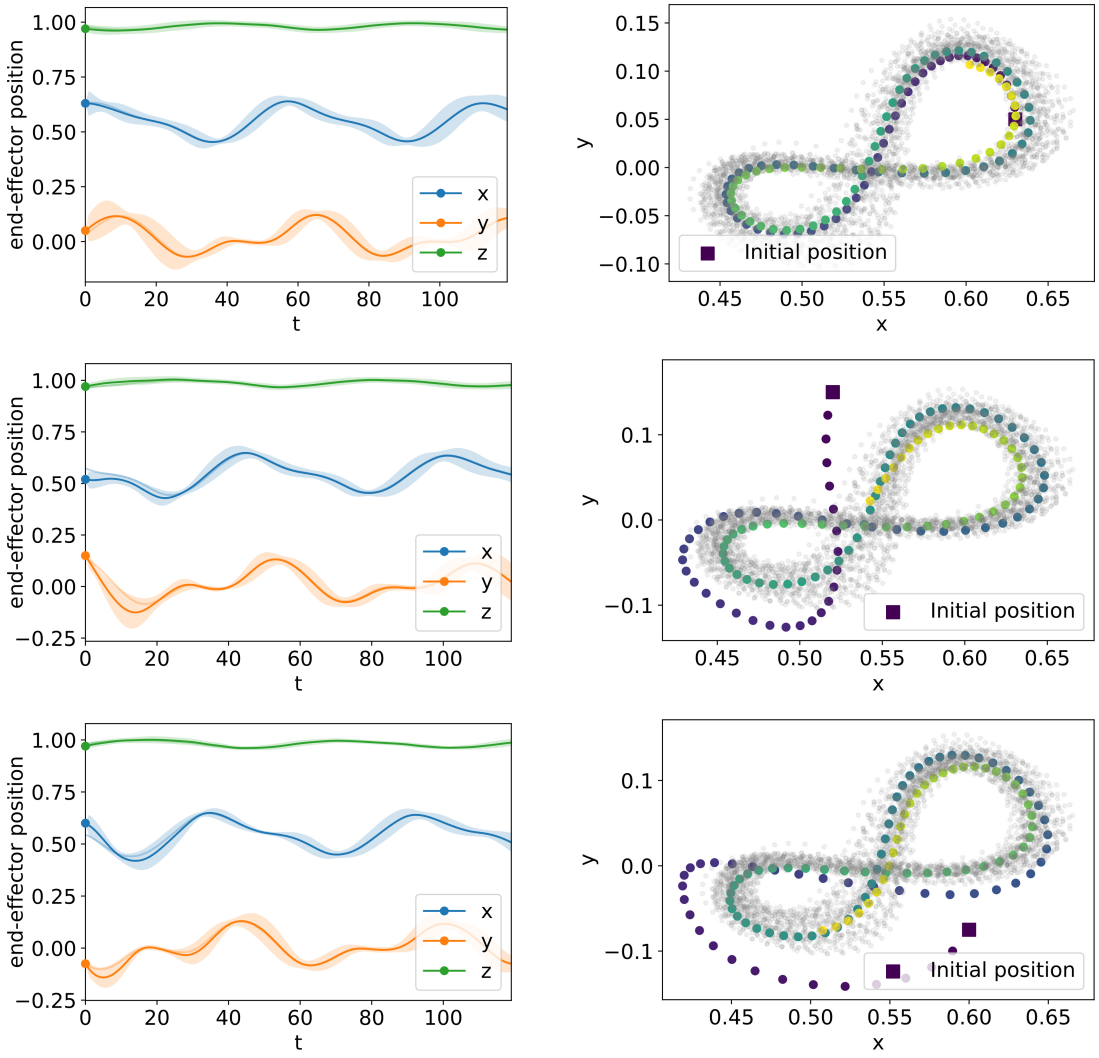
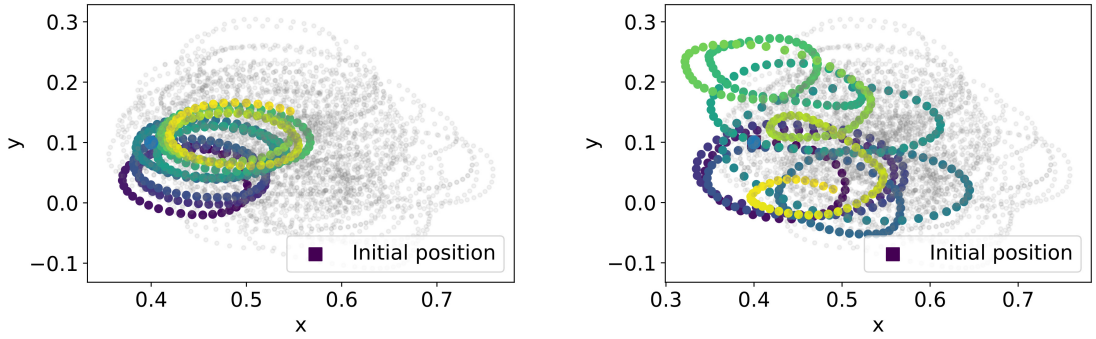


Figure 3.10 – 8-shape from different initial positions with FMP.



(a) Tracking the Gaussian mean

(b) Tracking a Gaussian sample

Figure 3.11 – Generated trajectories of length 400 (20s) for a given initial position.

We observe in Fig.3.11 that FMP is successful at generating trajectories of arbitrary lengths. In addition, sampling instead of tracking a Gaussian provides an interesting possibility, as we can see that the generated trajectory shows much more variability. This is useful for tasks that require some (co)variations in the movement (such as wiping tasks where we do not want artifacts to arise from a movement that repeats itself exactly).

3.6 Discussion

We now discuss the results from Section 3.5 and emphasize the advantages of FMP over other state-of-the-art methods.

We have shown that **FMP does not require demonstration alignment**, as it performs statistics directly over phase shifts in the complex weight space. It therefore goes beyond ProMP with cosine or Von-Mises basis functions, which fails when demonstrations are not aligned (see Fig.3.6). However, we have seen that increasing the number of Gaussians in ProMP can also permit to alleviate the need for demonstrations alignment. Moreover, when using ProMP-VM, only one frequency can be approximated, and it additionnally requires the extraction of this frequency as an external preprocessing step. FMP does not require such preprocessing.

FMP can learn tasks that involve a superposition of signals of different frequencies. This could not be done with the standard ProMP-VM. However, for a fairer comparison, we proposed to extend ProMP-VM to different frequencies by adding basis functions of different frequencies. We have shown that doing so permits to learn tasks that require different frequencies, but that the distribution learned is not as accurate as the one learned with FMP (we identified a factor 2 in terms of performance for our experiment, see Table 3.1). Also, FMP can represent variations of phase and amplitude for a given basis function in a single weight, by exploiting complex number properties. Furthermore, as many basis functions need to be placed for each frequency for ProMP-VM-Mult, this would not scale with the number of basis functions needed. We observed empirically that we could not include higher frequencies in ProMP-VM-Mult, as the redundancy and number of the basis functions led to numerical instabilities when learning the Gaussian

mixture. In contrast, FMP scales well with the number of basis functions, as we use all of them in our experiments. Better statistics might be obtained by performing dimensionality reduction of the number of basis functions and we plan to address this in future work.

Defining appropriate hyperparameters for basis functions in ProMP can be cumbersome. This holds true for ProMP-VM with periodic signals as well. Indeed, the number and centers of the basis functions need to be appropriately chosen (too few would make a very coarse discretization of the phase shifts, too many would lead to a very high number of basis functions, and hence poor statistics and/or numerical instabilities). **With FMP, no such choice is required**, as the complex exponentials form a basis and can approximate any signal. We therefore have a theoretical guarantee that demonstrations can be represented by weights.

In practice, one of the few hyperparameters that needs to be chosen with FMP is the length of the signal T to cut the demonstration(s). We noticed empirically that it had no effect on the final solution, as long as T is big enough to contain one or more periods. FMP does not require T to be set such that subdemonstrations are equal at the beginning and at the end. FMP just uses higher frequencies to compensate for this, but we did not observe any problem in our experiments.

We also showed that FMP has interesting extrapolation capabilities. While theoretically possible, conditioning to find a distribution that goes through a keypoint is not applicable in high dimensions, as it collapses to the mean of the distribution and hence does not go through the desired keypoint. We therefore proposed another way that is fast (solving of a least squares problem) and applicable to our high-dimensional setting (see Section 3.4.2). Additionnaly, we showed that it is safe when faced with a new situation (see Section 3.5.3). Not only does it return to the demonstrations, but it does so in a way that exploits the variations of magnitude and phase that were observed in the demonstrations. In the first two experiments (polishing and 8-shape), this means that the generated trajectories return back to the limit cycle. This is a property that is usually desirable for dynamical systems, which is not satisfied by ProMP-VM, as we saw in Fig.3.7. Moreover extrapolation with ProMP-VM might not be safe, as conditioning far from the Gaussian mean can result in overconfident trajectory predictions and hence highly stiff control around a potentially poor generalized trajectory.

3.7 Conclusion

We proposed a method based on discrete Fourier transform and Probabilistic Movement Primitives, which we call Fourier Movement Primitives (FMP) for the learning of rhythmic movements from demonstrations. Our basis functions are theoretically well motivated and no demonstrations alignment is required, which reduces the engineering burden. We have shown that FMP can learn tasks that involve a superposition of basis functions of different frequencies. The extrapolation capabilities of FMP are also relevant, generating trajectories that go back to the demonstrations when faced with a situation different from what was observed.

Future work will consider dimensionality reduction in the space of weights, which could enable the use of better control strategies in the Fourier domain, by using for example a Linear Quadratic Regulator (LQR). We will also study the possibility to perform statistics separately for the phase and magnitude of the weights, as it could yield richer compliance control strategies with an adaptive modulation of phase and amplitude.

We will also study the applicability of the approach for discrete (point-to-point) motions: for a series of demonstrations of the same length T we can symmetrize each demonstration, which would give periodic signals of length $2T$, and the method can be applied with no change. In the more general case where demonstrations do not have the same length, one possibility could be to symmetrize several times the shorter demonstrations to match the longest one, and performing statistics on those transformed demonstrations. Though the method would be directly applicable in this case, the performance and usefulness compared to other types of basis functions would have to be demonstrated.

4 Active Learning of Bayesian Probabilistic Movement Primitives

In the previous chapter, we have proposed an approach for learning robot skills from human demonstrations, with minimal user workload. Such learning from demonstration framework permits non-expert users to easily and intuitively reprogram robots. While such framework is particularly interesting for its adaptability to various situations where a robot engineer might not be available, its generalization and adaptation capabilities are heavily dependent on the quality and diversity of the demonstrations provided. We can give the following example to illustrate this point: let us consider a pouring task where we would like the robot to be able to adapt to different situations, such as the initial container being filled at different levels, or the final amount of liquid desired in the recipient. If a user only provides demonstrations that start with the container being full, this would result in very poor robot generalization capabilities when the container is not initially full, leading to task failure. Unfortunately, providing or requesting good demonstrations is not easy, as quantifying what constitutes a good demonstration in terms of generalization capabilities is not trivial.

In this chapter, we propose an active learning method for contextual probabilistic movement primitives 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.

4.1 Introduction

Learning from demonstration (LfD) offers an intuitive framework for non-expert users to easily (re)program robots. As discussed in Chapter 2, one well-established LfD approach is to learn probability distributions over trajectories, for example using the framework of ProMPs. One of the main capabilities of ProMPs lies in the task generalization, which is usually achieved by

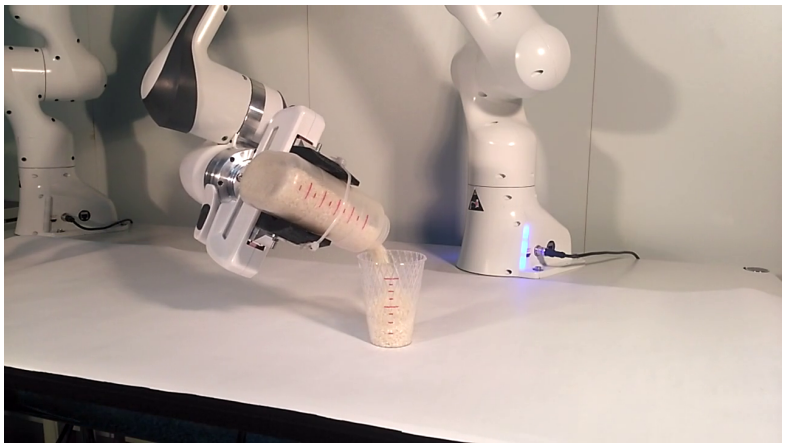


Figure 4.1 – Overview of the pouring task with a 7-axis robot.

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 [42, 106]. 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 [126]. 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 [125]. 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.

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 [127]. 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 chapter, 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 [15] to quantify both aleatoric and epistemic uncertainties in ProMPs. Specifically, we propose to learn the ProMP with a Bayesian Gaussian mixture model (BGMM). In Sec. 4.3, we introduce Bayesian ProMPs. Then, in Sec. 4.4, we propose an active learning method based on the epistemic uncertainties captured by the BGMM. We demonstrate the applicability of our approach in Sec. 4.6 on four different pouring task experiments. The first three 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 chapter 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.

4.2 Related work

In this section we build upon 2.1.1 and review more specifically the literature that has considered active learning for imitation learning. Indeed, as the data acquisition process is usually costly in robotics, active learning has emerged as a viable solution [119, 96, 122, 129]. It has been shown that active learning permits a faster exploration of the action space, which is particularly true in the context of developmental robotics, where active learning is often referred to as curiosity-driven learning (see 2.1.1). In the context of learning from demonstrations, active imitation learning is a topic gaining interest. It has indeed been successfully used in a variety of robotic tasks, such as autonomous navigation [131, 78]. In [25], 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 [129, 27]. In [55], 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 [87], 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 [29] 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 [127] 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.

4.3 Bayesian ProMPs

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

4.3.1 Contextual ProMP

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 [105, 42] to take into account context variables is to learn the joint distribution of contexts and ProMP 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 $p(\boldsymbol{c}, \boldsymbol{w}) = p(\tilde{\boldsymbol{w}})$.

4.3.2 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 $p^{\mathrm{GT}}(\boldsymbol{c}, \boldsymbol{w})$ that can be used to generate robot movements $p^{\mathrm{GT}}(\boldsymbol{w}|\boldsymbol{c})$ adapted for context $\boldsymbol{c}$. We aim to learn this unknown joint distribution by active imitation learning.

4.3.3 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:

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k|(\beta_0\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{S}_k, \nu_k), \tag{4.1}$$

$$p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\alpha_0). \tag{4.2}$$

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 Alg.2 in Chapter 2 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 $\hat{\nu}_k$

$$\begin{aligned} p(\hat{\tilde{\boldsymbol{w}}}|\tilde{\boldsymbol{W}}) = \sum_{k=1}^{K} \hat{\pi}_k \, \mathrm{t}(\hat{\tilde{\boldsymbol{w}}}|\hat{\boldsymbol{m}}_k, \hat{\boldsymbol{L}}_k, \hat{\nu}_k), \qquad & \text{where} \\ \hat{\pi}_k &= \frac{\alpha_k}{\sum_{k=1}^K \alpha_k}, \\ \hat{\nu}_k &= \nu_k + 1 - D - D^c, \\ \hat{\boldsymbol{L}}_k &= \frac{(\nu_k + 1 - D - D^c)\beta_k}{1+\beta_k}\boldsymbol{S}_k, \\ \hat{\boldsymbol{m}}_k &= \boldsymbol{m}_k, \end{aligned} \tag{4.3}$$

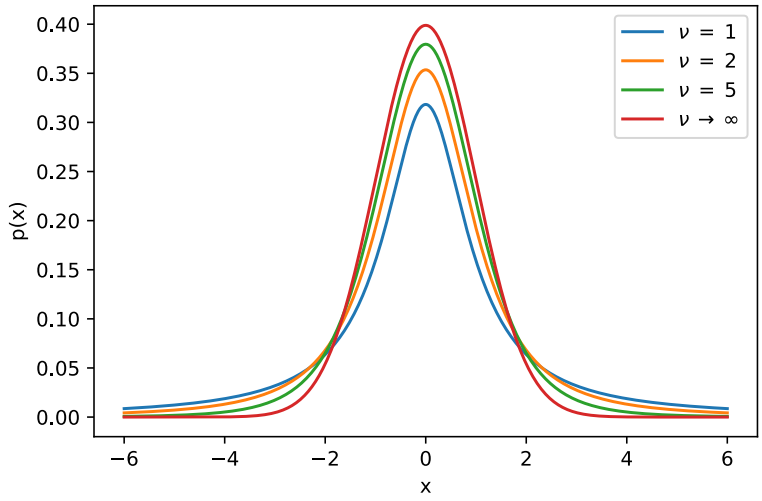


Figure 4.2 – Probability density functions of univariate t-distributions for several degrees of freedom.

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 Alg.2 for more details. A multivariate t-distribution is very similar to a multivariate Gaussian distribution, but its tails decay more slowly. A Gaussian distribution is actually a special case of a t-distribution when the number of degrees of freedom tends to infinity. We provide in Fig.4.2 a visualization of univariate t-distributions for different degrees of freedom.

We can then condition on the context to get the conditional posterior predictive distribution of the weights for a given context variable as in [15] (Section 10.2.3)

$$p(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}}) = \sum_{k=1}^{K} \hat{\pi}_k^{w|c}\, \mathrm{t}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{m}}_k^{w|c}, \hat{\boldsymbol{L}}_k^{w|c}, \hat{\nu}_k^{w|c}), \tag{4.4}$$

$$\text{with} \quad \hat{\pi}_k^{w|c} = \frac{\hat{\pi}_k\, \mathrm{t}(\hat{\boldsymbol{c}}|\hat{\boldsymbol{m}}_k^{c}, \hat{\boldsymbol{L}}_k^{c}, \nu_k^{c})}{\sum_{j=1}^{K} \hat{\pi}_j\, \mathrm{t}(\hat{\boldsymbol{c}}|\hat{\boldsymbol{m}}_j^{c}, \hat{\boldsymbol{L}}_j^{c}, \nu_j^{c})}, \tag{4.5}$$

$$\hat{\nu}_k^{w|c} = \hat{\nu}_k + D^c, \tag{4.6}$$

$$\hat{\boldsymbol{m}}_k^{w|c} = \hat{\boldsymbol{m}}_k^{w} + \hat{\boldsymbol{L}}_k^{wc}\hat{\boldsymbol{L}}_k^{cc^{-1}}(\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^{c}), \tag{4.7}$$

$$\hat{\boldsymbol{L}}_k^{w|c} = \frac{\hat{\nu}_k + (\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^{c})^\top \hat{\boldsymbol{L}}_k^{cc^{-1}}(\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^{c})}{\hat{\nu}_k^{w|c}} \cdot (\hat{\boldsymbol{L}}_k^{ww} - \hat{\boldsymbol{L}}_k^{wc}\hat{\boldsymbol{L}}_k^{cc^{-1}}\hat{\boldsymbol{L}}_k^{wc^\top}), \tag{4.8}$$

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.

4.4 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.

4.4.1 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. (4.8), we can see that the covariance matrix of the conditional posterior predictive distribution can be decomposed into two parts (see also [55])

$$\hat{\boldsymbol{L}}_k^{w|c} = \hat{\boldsymbol{L}}_k^{\mathrm{al}} + \hat{\boldsymbol{L}}_k^{\mathrm{ep}}, \qquad \text{where} \tag{4.9}$$

$$\hat{\boldsymbol{L}}_k^{\mathrm{al}} = \frac{\hat{\nu}_k}{\hat{\nu}_k^{w|c}} (\hat{\boldsymbol{L}}_k^{ww} - \hat{\boldsymbol{L}}_k^{wc} \hat{\boldsymbol{L}}_k^{cc^{-1}} \hat{\boldsymbol{L}}_k^{wc^\top}), \tag{4.10}$$

$$\hat{\boldsymbol{L}}_k^{\mathrm{ep}} = \frac{(\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^c)^\top \hat{\boldsymbol{L}}_k^{cc^{-1}} (\hat{\boldsymbol{c}} - \hat{\boldsymbol{m}}_k^c)}{\hat{\nu}_k^{w|c}} (\hat{\boldsymbol{L}}_k^{ww} - \hat{\boldsymbol{L}}_k^{wc} \hat{\boldsymbol{L}}_k^{cc^{-1}} \hat{\boldsymbol{L}}_k^{wc^\top}). \tag{4.11}$$

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 [15] (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.

4.4.2 Uncertainty measurement

The most general and common uncertainty measure is the Shannon entropy [128]. 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:

$$\tilde{\pi}_k(\boldsymbol{c}) = \hat{\pi}_k^{w|c}, \qquad \tilde{\boldsymbol{\mu}}_k(\boldsymbol{c}) = \hat{\boldsymbol{m}}_k^{w|c}, \qquad \tilde{\boldsymbol{\Sigma}}_k(\boldsymbol{c}) = \frac{\hat{\nu}_k^{w|c}}{\hat{\nu}_k^{w|c} - 2} \hat{\boldsymbol{L}}_k^{\text{ep}}(\boldsymbol{c}). \tag{4.12}$$

We propose to use the closed-form lower bound introduced in [75], which has been shown to be tight. It is expressed as (for clarity purposes we omit the fact that all GMM parameters depend on $\boldsymbol{c}$)

$$H_{lower}\big(p^{ep}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}})\big) = \frac{1}{2}\Big(K \log 2\pi + K + \sum_{i=1}^{K} \tilde{\pi}_i \log |\tilde{\boldsymbol{\Sigma}}_i|\Big) \\ - \sum_{i=1}^{K} \tilde{\pi}_i \log \sum_{j=1}^{K} \tilde{\pi}_j e^{-C_\alpha(p_i, p_j)}, \tag{4.13}$$

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

$$C_\alpha(p_i, p_j) = \frac{(1-\alpha)\alpha}{2} \cdot (\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j)^\top \Big((1-\alpha)\tilde{\boldsymbol{\Sigma}}_i + \alpha\tilde{\boldsymbol{\Sigma}}_j\Big)^{-1} (\tilde{\boldsymbol{\mu}}_i - \tilde{\boldsymbol{\mu}}_j) \quad + \\ \frac{1}{2} \log \left(\frac{|(1-\alpha)\tilde{\boldsymbol{\Sigma}}_i + \alpha\tilde{\boldsymbol{\Sigma}}_j|}{|\tilde{\boldsymbol{\Sigma}}_i|^{1-\alpha} |\tilde{\boldsymbol{\Sigma}}_j|^{\alpha}} \right). \tag{4.14}$$

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 5. 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.

4.5 Illustrative examples

In this section, we provide more details on our motivation for considering epistemic uncertainties for active learning. First, we motivate it with a simple coin tossing example where we demonstrate that separating different sources of uncertainties is necessary for active learning. Then, we illustrate aleatoric and epistemic uncertainties on a 2D toy problem to give the reader an intuitive understanding about those concepts. Finally, we highlight the difference between our approach

Algorithm 5: 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 (4.4) to (4.8);
Isolate the epistemic uncertainty $p^{ep}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}})$ with Equations (4.9) and (4.11);
Approximate the entropy of $p^{ep}(\hat{\boldsymbol{w}}|\hat{\boldsymbol{c}}, \tilde{\boldsymbol{W}})$ with Equations (4.12) to (4.14);

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

and the more commonly used Gaussian Processes using the latter 2D toy problem.

4.5.1 Why epistemic uncertainties?

We present here a simple coin tossing problem to illustrate the need to separate the aleatoric and epistemic uncertainties. Let us imagine we have two coins:

- Gold coin: this coin is not tricked and there is exactly a 0.5 probability of getting heads or tail.

- Silver coin: we know this coin is tricked and there is a higher probability of getting heads than tail. For simplifying the example, let us consider that we know that the probability of getting heads is 0.6 or 0.7, but we do not know which value [I]. We suppose we have the same belief over whether the probability is 0.6 or 0.7.

Similarly to our robotics application, our goal is to learn the conditional model $p(\textit{heads}|\textit{coin chosen})$. In other words, our goal is to learn to predict the environment as best as we can. Given this goal, which coin should we decide to toss? It seems obvious that we should toss the silver coin, because we have an uncertainty whether its probability of falling on heads is 0.6 or 0.7, whereas we already know that the gold coin is not tricked. We calculate now the entropies associated to the choice of each coin:

[I] Note that the following reasoning would also be valid given just the belief that the coin is tricked and no additional assumptions, we aim here to simplify the example and the calculus.

$$H(\textit{Gold}) = -\sum_{p\in[p_{\text{heads}},p_{\text{tail}}]} p\log p = -(0.5\log 0.5 + 0.5\log 0.5) = -\log 0.5 \approx \mathbf{0.69}$$

$$H(\textit{Silver}) = -\sum_{p\in[p^1_{\text{heads}},p^1_{\text{tail}},p^2_{\text{heads}},p^2_{\text{tail}}]} = -0.5\left(\sum_{p\in[p^1_{\text{heads}},p^1_{\text{tail}}]} p\log p\right) - 0.5\left(\sum_{p\in[p^2_{\text{heads}},p^2_{\text{tail}}]} p\log p\right)$$

$$= -0.5\ (0.6\log 0.6 + 0.4\log 0.4) - 0.5\ (0.7\log 0.7 + 0.3\log 0.3) \approx \mathbf{0.64}$$

We can see that $H(\textit{Gold}) > H(\textit{Silver})$, so if we were to choose the coin for which we have the more uncertainties about the outcome, we would choose the gold coin. Indeed, we have a belief that the silver coin returns more frequently heads than tails, so we have more information about the outcome of tossing the silver coin than the gold coin for which we know that it return heads with probability 0.5.

This example illustrates that more uncertainties do not necessarily reflect more to learn. There are indeed two types of uncertainties: uncertainty related to noise about which we can't do anything, no matter how many samples we observe, and uncertainty related to our lack of knowledge of the environment. In this example, there is great uncertainty related to choosing the gold coin, but this is noise and tossing the gold coin does not permit to improve our prediction. This is the motivation for separating those two types of uncertainties for our active learning method: we want to avoid the robot exploring environment noise from which there is nothing to be learned.

It is important to note here that the notions of aleatoric (noise) and epistemic (lack of knowledge) uncertainties are not absolute notions [63]. They refer to uncertainty that is non-reducible or reducible, but it may not be trivial to separate them depending on the machine learning model used and of the application. The Bayesian Gaussian mixture model we use lends itself very well to this distinction but this is not the case for all models. For instance, even on our simple coin tossing example, it is not trivial to separate the entropy related to choosing the silver coin between two terms that could be attributed to reducible and non-reducible uncertainties.

4.5.2 Visualization of uncertainties

In this subsection, we provide more insights about those notions of aleatoric and epistemic uncertainties with a visualization of those on a two-dimensional toy problem.

We show in Fig.4.3a a toy dataset generated for illustration purposes. The underlying conditional model that we want to learn is here a very simple identity function. But, this function exhibits different noises for different regions of the input space. We suppose that we have some data in two regions of the input space, for the left one the noise is high, and for the right one the noise is

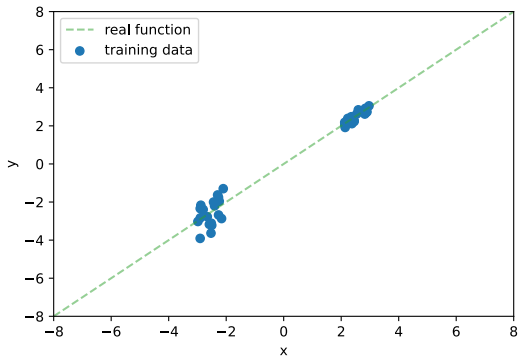
low. We will illustrate the different uncertainties and entropies on this visual two-dimensional example. We learn a Bayesian Gaussian mixture model with two Gaussians on this data, and plot the *maximum a posteriori* in Fig.4.3b. We can see that the result is very similar to the standard maximum likelihood Gaussian mixture model. [II]

We can now calculate the conditional distribution using the learned BGMM, and separate its covariance matrix into the aleatoric and epistemic terms (see Equations (4.4) to (4.11)). We show in Fig.4.3c the aleatoric uncertainty. We observe that it is constant in the left part of the input space, with a significant value of the uncertainty, and that it is also constant in the right part of the input space, with a very small uncertainty value. We can note here that the aleatoric uncertainty does not depend on the input on which we condition, given that we stay in a given cluster (Gaussian). We can also see that the value of the aleatoric uncertainty is actually equal to the noise in each cluster of the data. It is important to note here that this is the type of uncertainty that we get with a standard maximum likelihood (non-Bayesian) Gaussian mixture model, and it is the reason why they are known to predict overconfident predictions far from the training data [9].

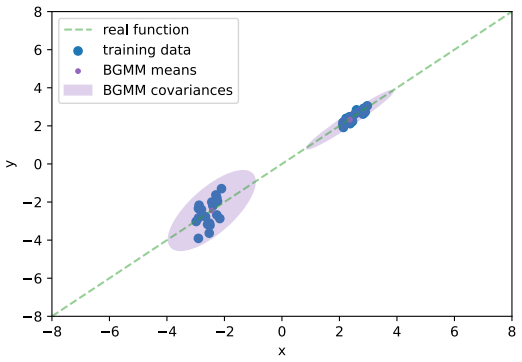
In Fig.4.3d, we plot the value of the epistemic uncertainty on the same data. We observe that it is equal to zero close to the training data, no matter the level of noise. Also, we can see that it grows quadratically as we try to make predictions far from the training data. It is interesting to note that the epistemic uncertainties grow faster around the region of high noise than around the region of low noise. This is because the noise of the cluster influences the value of the epistemic uncertainty as a multiplicative factor (see Eq.(4.11)).

Given those observations, we can get the intuition of why considering aleatoric uncertainties for active learning would result in very poor learning, and the necessity to consider the epistemic uncertainties that are related to the lack of knowledge and not to the level of noise. We also plot the entropies of the conditional distribution in Fig.4.3e, which are calculated as proposed in 4.4.2. This highlights that if we were to choose active learning queries based on the entropy of the aleatoric uncertainty, we would only explore noisy regions of the input space. It also shows that the entropy of the epistemic uncertainty is high in three regions of the input space, which are those that are far from the training data. We also plot here the entropy of the full conditional distribution, i.e., where the covariances are the sums of the aleatoric and epistemic terms. This illustrates the need for separating the uncertainties for active learning instead of using the full (aleatoric plus epistemic) conditional distribution: the input $x = -2.5$ in the middle of the noisy training data has a total entropy of about 1, while the input $x = 8$ has a total entropy of about 0. This suggests that the noise part is an important part of the conditional distribution, and that the total entropy would be very biased towards exploring noisy regions.

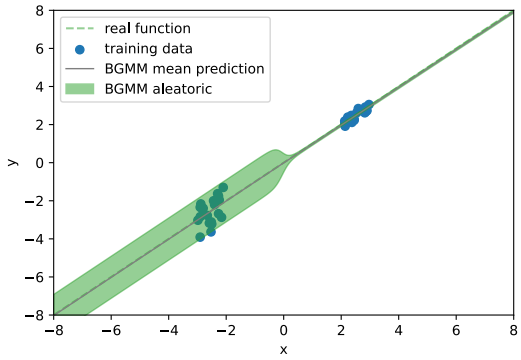
[II] Note that different choices of priors can change the joint distribution learned with the BGMM. For instance, a high mean concentration prior would tend to bring the two Gaussians means closer to the mean prior, which is usually chosen as the mean of the data (here, $(0, 0)$). Also, for visualization purposes we choose a number of Gaussians which is equal to the number of clusters in the data. In the case where more Gaussians would be considered, two Gaussians would fit the clusters as in Fig.4.3b, and the others would *fall* into the prior, i.e., have the mean prior and covariance prior.



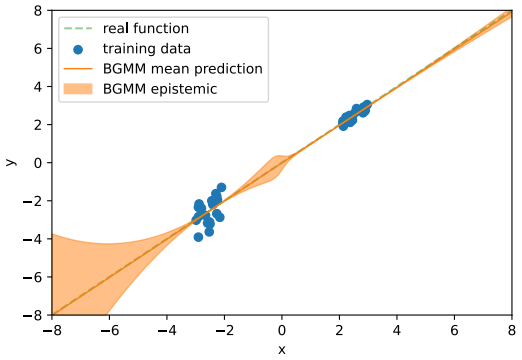
(a) Toy data



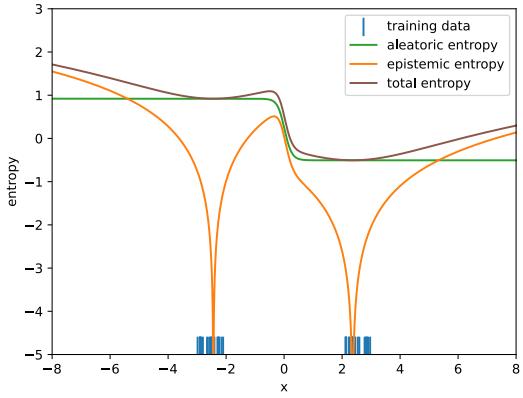
(b) Joint distribution learned



(c) Aleatoric uncertainty



(d) Epistemic uncertainty



(e) Entropy of the conditional distribution

Figure 4.3 – Visualization of the aleatoric and epistemic uncertainties of a Bayesian Gaussian mixture model

4.5.3 Why not Gaussian Processes?

We illustrate here the difference between our approach and the more commonly used Gaussian Processes (GPs) for active learning. The main drawback of Gaussian Processes is its assumption of homoscedastic noise, i.e., the assumption that the noise value is the same in the whole input space. We show in Fig.4.4 visualizations of learned GPs on the 2D toy problem studied in previous subsection[III]. We notably highlight two different choices of noise: a small value (as in the right data cluster), and a high value (as in the left data cluster). We can see that GPs cannot model the different levels of noises of the training data because of the homoscedasticity assumption. If the noise level is chosen low, the data is fitted quite well but the prediction of the variance for the noisy cluster is way too low, meaning overconfident predictions. In the context of robotics, this might mean that the variance predicted does not reflect the demonstrated variance, and hence cannot be used reliably, for instance for compliance control. If the noise level is chosen high, then the fitting of the non-noisy data cluster is very bad, and its associated variance is high. In robotics, this can mean that if demonstrations over a certain region of the input space are highly precise and do not exhibit variations, the robot would still allow itself some variability around those demonstrations, and hence would not have captured the essence of the movement.

It is worth noting that though noise homoscedasticity is a very common assumption in GPs, there are works that have proposed ways to alleviate it to consider heteroscedastic noise [80, 79, 67]. This usually comes at the expense of a higher computational cost.

Another significant difference between GPs and BGMMs is that GPs model the conditional distribution, while BGMMs model the joint distribution. Thus, it is possible to extract several conditional distributions from the joint distribution learned with a BGMM. We will see in the next chapter that it can be useful for considering several learning modalities.

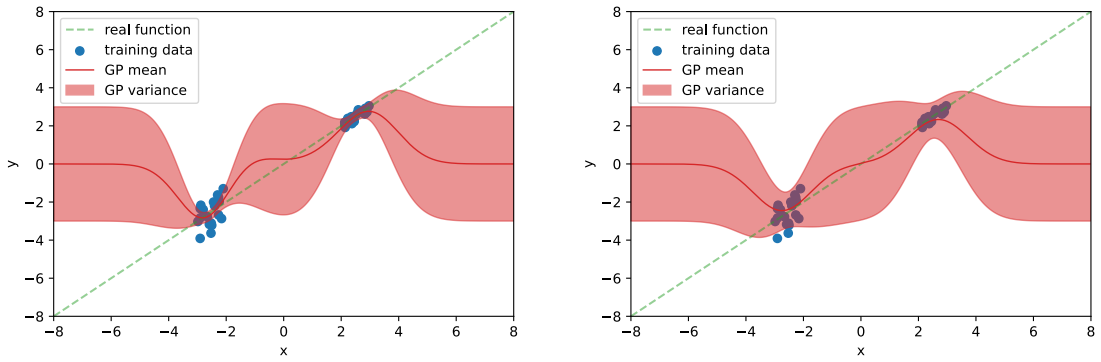
4.6 Experiments

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)[IV] for ProMP. The width of the RBFs are set as $h = (\frac{T-1}{N})^2$. The hyperparameters of the BGMM are

[III] Note that we used here the null function $\mu(x) = 0$ as mean prior, which is why the model predicts 0 far from the training data. If we had a prior knowledge that the data is linear, it would be possible to choose the mean prior as $\mu(x) = x$ instead, which would result in the model predicting the identity far from the training data. Given that we did not use such prior knowledge for the BGMM, for a fair comparison we also do not use it here.

[IV] Note that we could alternatively use the Fourier basis functions used in the previous chapter. We chose not to since this would come at a greater computational complexity as the number of the Fourier basis functions is considerably bigger, for a gain that would probably be minor because the demonstrations considered do not involve strong misalignments.



(a) Small homoscedastic noise

(b) High homoscedastic noise

Figure 4.4 – Visualization of the uncertainties of a Gaussian Process

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 [87] for external context variables: we learn the conditional model of the trajectories given the context with a Gaussian process (GP)[V] 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 [29] (introduced in Sec. 4.2) 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 [29] for more details[VI].

4.6.1 Simulated pouring

We use here a simulated pouring environment implementing the Franka Emika robot in the PyBullet simulator [30]. 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. 4.5. In the first two simulated environments, we avoid learning the affordances of the object and control directly the orientation of the edge of

[V] 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 thesis.

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

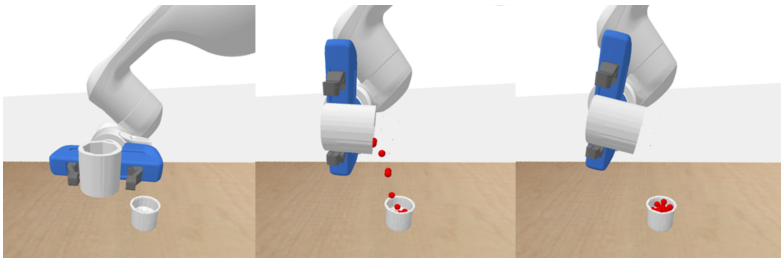


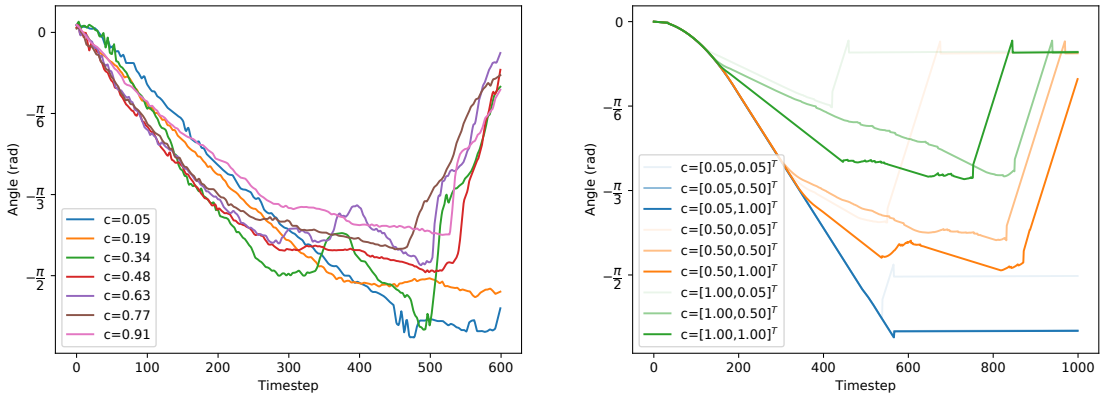
Figure 4.5 – Overview of the simulated pouring environment.

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.

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).

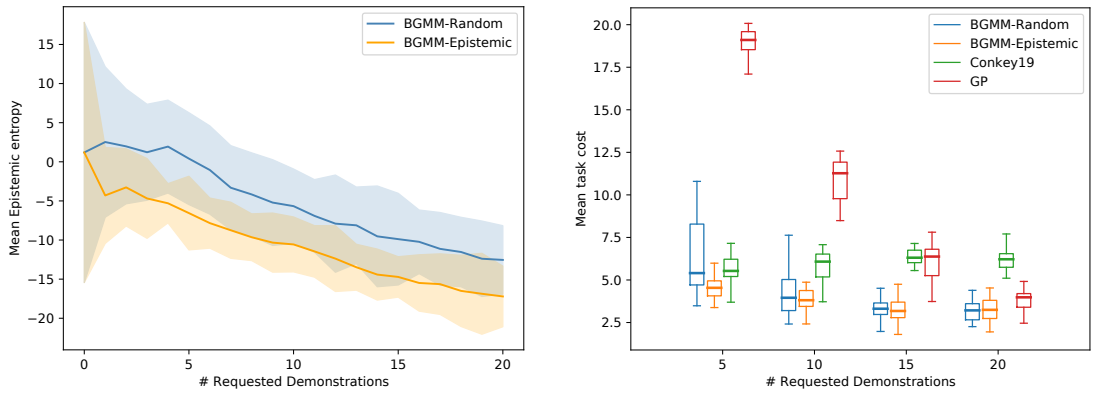
In order to have demonstrations exhibiting realistic variations, we provide real human demonstrations using teleoperation. As the reference trajectory contains only a one-dimensional angle, teleoperation is made simply using a camera by detecting the angle of a colored object held by the human demonstrator. 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. 4.6a. 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. 4.7 how it compares to a random strategy which randomly chooses the next context. In Fig. 4.7a, 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. 4.7b, 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



(a) Teleoperated demonstrations, 1D context

(b) Generated demonstrations, 2D context

Figure 4.6 – Subset of demonstrations for different contexts.



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

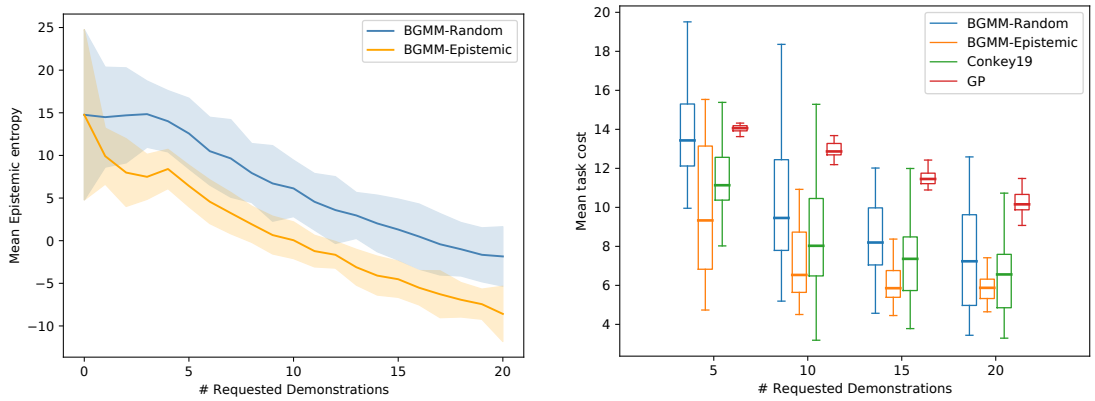
(b) Reduction of task cost w.r.t number of demonstrations

Figure 4.7 – Quantitative results for simulated 1D context pouring.

being almost completely filled. Filling it too much is possible and increases the task cost as well). We observe in Fig. 4.7b 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.

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. This means that all contexts may not be feasible, and that the user would just provide the demonstration that is the closest from the robot request. Alternatively, one could define more precisely the context space so that all contexts are feasible). A sample of generated demonstrations can be found in Fig. 4.6b. 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. 4.8. We can see in Fig. 4.8a that our strategy outperforms the random strategy in terms of reduction of the epistemic uncertainties. More importantly, we see in Fig. 4.8b 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



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

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

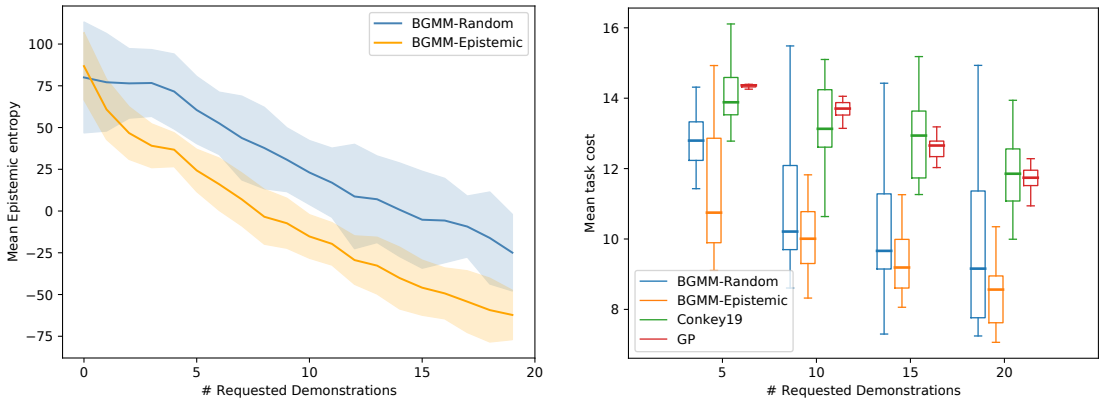
Figure 4.8 – Quantitative results for simulated 2D context pouring.

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.

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 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 [10] implemented in the *hyperopt* Python package [11]) to calculate the context yielding the highest epistemic entropy.

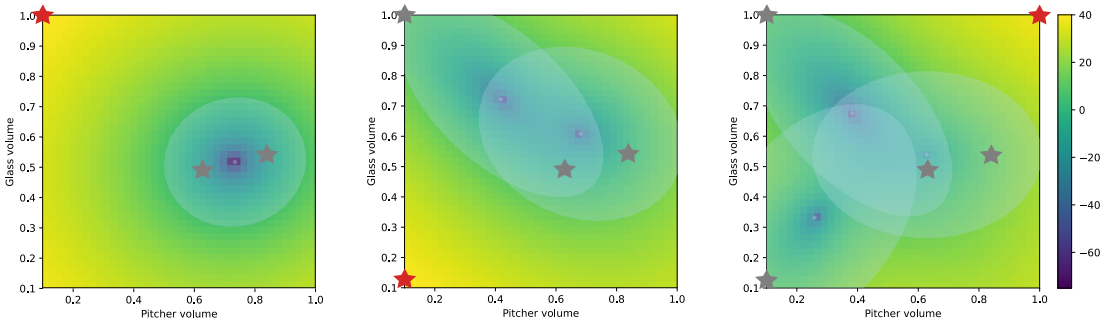
We can see in Fig. 4.9a that the reduction of the epistemic uncertainties is bigger with our active learning metric than with the random baseline, similarly to what we observed in the past



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

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

Figure 4.9 – Quantitative results for simulated 3D context pouring.



(a) First iteration

(b) Second iteration

(c) Third iteration

Figure 4.10 – 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.

two experiments, and that this epistemic reduction correlates with a better task cost error (see Fig. 4.9b), 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.

4.6.2 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. 4.1. 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. 4.10. 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[VII] (we tested it on 9 different contexts, taken from a 3×3 grid in the context space).

4.7 Conclusion

In this chapter, 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.

[VII] https://sites.google.com/view/bayesianpromps

5 Combining Social and Intrinsically-Motivated Learning for Multi-Task Robot Skill Acquisition

In the previous chapter, we proposed an approach for active learning from demonstrations. We showed that active learning permitted to reduce the overall number of demonstrations required to learn a task, while requiring fewer demonstrations. Such learning framework is still dependent on a single learning modality: learning from demonstrations. We believe that this might limit the possible range of applications, since some tasks might be simply too complex to be learned purely from demonstrations. Examples can be tasks where humans cannot provide accurate demonstrations, or tasks inherently too complex that would require too many demonstrations to be learned properly. In this chapter, we build upon previous chapter and propose an approach for coupling internally-guided learning and social interaction in the context of a multi-task robot skill acquisition framework. More specifically, we focus on learning a parametrized distribution of robot movement primitives by combining active intrinsically-motivated learning and active imitation learning. We focus on the case where the learning modalities to use are not specified in advance by the experimenter, but are chosen actively by the robot through experiences. Such approach aims at combining experiential and observational learning as efficiently as possible, by relying on a skill acquisition mechanism in which the agent/robot can orchestrate different learning strategies in an iterative manner, and modulate the use of these modalities based on previous experiences. We demonstrate the effectiveness of our approach on a waste throwing task with a simulated 7-DoF Franka Emika robot, where at each iteration of the learning process the robot can actively choose between observational/imitation learning and experiential/intrinsically-motivated learning.

5.1 Introduction

Humans and other animals acquire and refine skills in an open-ended manner through lifelong learning, and are hence autonomous and versatile for interacting and learning in their environments. Despite the important progress in Artificial Intelligence, robots still lack this capacity. Endowing robots with the capability to autonomously discover and solve multiple tasks incrementally and in an open-ended manner is one of the greatest challenges of robotics today and

is the goal of the growing field of developmental robotics [85]. In particular, humans have the ability to use several learning modalities, and most interestingly to arbitrate their choice based on their reliability [26, 59, 142]. In this chapter, we explore a possible route towards such a goal by proposing a principled computational approach combining intrinsically-motivated learning and imitation learning.

In robotics, skills acquisition is most often studied by concentrating on a single learning strategy, or by predefining a basic sequence of learning strategies in advance (e.g., a reinforcement learning problem initialized with a demonstration). This led to large research efforts dedicated to the development of very elaborated algorithms specialized in a single domain (learning from demonstration, reinforcement learning, curiosity-driven learning). We argue that this complexity could be reduced by allowing several learning strategies, and by providing a mechanism to select these learning modalities in an open-ended and interactive manner. In the same way as we cannot learn to play football only by watching TV and that we cannot learn football tactics from scratch only based on the rules of the game, we believe that robots should rely on multiple learning strategies, whose sequence can only be determined during the course of learning, in a lifelong learning fashion.

The above argument is motivated by studies in various fields including cognitive science [54, 85], ethology [59, 142], neurocomputing [145, 26] and robotics [117, 121, 18, 136, 17], all proving insights, in different forms, about the importance of combining multiple learning modalities to acquire skills. In particular, several developmental studies such as [59, 142, 58] have shown that learning by imitation is a key component of social learning in child development. Children tend to imitate what they are shown, even if some of the observed actions are not necessarily useful.

From a developmental robotics point of view, we argue that orchestrating multiple learning strategies during the skill acquisition process can better cope with the specific advantages and limitations of each individual strategy. Indeed, these strategies are often complementary to each other, hence the necessity to combine them. Intrinsically-motivated learning requires no external guidance, i.e., no presence of a human, but it usually involves a long interaction process with the environment. Imitation learning, on the other hand, requires the presence of a human, but demonstrations provide a lot of information which would have required a tremendous amount of time to autonomously acquire.

In this chapter, we propose an active learning approach that can act on different fronts: at a meta-level, by deciding about the currently most appropriate learning modality in an open-ended manner, and at a low-level, by deciding about which of the condition/situation/context the agent currently needs to experience on its own or request as demonstration.

Our contribution is a Bayesian computational framework for learning robot movement primitives providing this high-level and low-level arbitration capability, namely:

- Strategy selection: the robot chooses actively between imitation learning and intrinsically-

motivated learning, based on its previous experiences.

- Demonstration choice: in the imitation learning strategy, the robot chooses actively the goal that is expected to yield the most interesting demonstration.

- Policy exploration: in the intrinsically-motivated learning strategy, the robot chooses actively which movement is going to improve the most its knowledge of the task.

To the best of our knowledge, our work is the first to integrate these three learning aspects in a computational framework.

This chapter is organized as follows. First, we review the existing literature in Sec. 5.2. In Sec. 5.3, we introduce our Bayesian computational framework, and in Sec. 5.4, we derive two active learning strategies as well as an arbitration strategy. Our experimental results are presented in Sec. 5.5.

5.2 Related Work

We focus here on the works that specifically combine intrinsically-motivated learning and social learning. The reader can refer to 2.1.3 for pointers to intrinsically-motivated works, and to 4.2 for the related work on active imitation learning.

Psychologists have observed on a tool use task that intrinsically-motivated learning can be more efficient if children can see an agent solve the task [58]. This suggests that a learning robot could benefit from combining intrinsically-motivated learning and social learning (e.g., imitation), instead of acquiring skills with a single learning modality. Several works in developmental robotics have indeed studied methods combining those modalities. In [95], Nguyen *et al.* propose an algorithm for combining intrinsically-motivated self-exploration and imitation learning. In particular, a solution is proposed to the problem of choosing what learning strategy is the most appropriate. In the context of a throwing task, they show that there is a significant gain in combining several learning strategies and actively choosing between them. Besides the fact that their method was only evaluated on a one degree of freedom robot, there is a fundamental difference between their approach and ours. They base the choice of their learning strategy on values of interest levels, which are computed with the progress previously observed when choosing the different modalities. This supposes a notion of competence (reward) to choose between the modalities. In contrast, our work bases its strategy selection process on uncertainties that are computed with a statistical model representing the data (intrinsically-motivated trials and demonstrations), and hence does not require the notion of an external reward. Additionally, the computation of the interest values in [95] requires the evaluation of the competence before and after each episode, which implies executing a large number of movements to measure the mean distance to the goal. Our method is based on an internal reward related to intrinsic motivation and alleviates therefore this limitation.

An extension of [95] relied on the use of procedures to combine primitive policies [38], and it has been applied on a real robot in [39]. Those approaches differ from our approach by the way they arbitrate between the learning modalities: they seek to choose the modalities based on the competence improvement they entailed in previous iterations. This implies that a learning modality that led high improvement will be more likely to be selected in the future. This also implies that all of the learning modalities have to be tried in different parts of the goal space in order to quantify appropriately their potential competence improvement in those different regions. This notably explains why those approaches rely on very long interaction processes, typically of serveral thousands of iterations. In contrast, our approach does not need to try out the different learning modalities to quantify their improvement, as the notion of improvement is based directly on the learned model of the movement, and targets small data applications, typically around 20 iterations. We will see that with our method the robot can know it is better to imitate in the beginning of the learning process than trying out by itself, even though it never tried the intrinsically-motivated learning strategy. An interest model for goal babbling is also used in [97], by relying on an external reward. In this work, Nguyen *et al.* show that social learning through human demonstrations can bootstrap the performance of an intrinsically motivated robot learner. In a simulated fishing task experiment in which the robot needs to learn how to reach various goals with a fishing rod, a demonstration is given at constant frequency, chosen randomly from the set of goals. They show that this permits to reduce the task cost compared to a purely intrinsically-motivated learning framework. As mentioned in the conclusions of the above papers, an interesting improvement would be to have the possibility to interactively choose the switching between those modalities. Our approach proposes a possible solution to this problem.

5.3 Bayesian Movement Representation

We use the movement representation presented in previous chapter, which is a Bayesian extension of the widely used of probabilistic primitives. We review it briefly:

- Demonstrations are mapped to a lower-dimensional space (ProMP weight space) using basis functions
- The joint distribution of the demonstrations is learned in the weight space using a Bayesian Gaussian Mixture Model
- When conditioning, we separate the two types of uncertainties: aleatoric and epistemic, and quantify the epistemic uncertainties using a closed-form lower bound of the Shannon entropy for Gaussian mixtures.

We refer the reader to Section 4.3 for more details on the BGMM learning and conditioning, and to Section 4.4 for the uncertainty decomposition and measurement.

In contrast to previous chapter, we will consider here task adaptation with respect to a trajectory

via-point (final location) instead of an external context space. Due to the linear relation from trajectory space to weight space inherited from ProMPs, note that it is possible to condition on a trajectory via-point/s $\hat{\boldsymbol{\tau}}^c$ directly to get $p(\hat{\boldsymbol{w}}|\hat{\boldsymbol{\tau}}^c, \tilde{\boldsymbol{W}})$, with minimal changes compared to the previously considered context conditioning case: this is done simply by replacing all $\hat{\boldsymbol{m}}_{\boldsymbol{k}}^*$ and $\hat{\boldsymbol{L}}_{\boldsymbol{k}}^*$ in (4.4)–(4.8) by $\boldsymbol{\Phi}\hat{\boldsymbol{m}}_{\boldsymbol{k}}^*$ and $\boldsymbol{\Phi}\hat{\boldsymbol{L}}_{\boldsymbol{k}}^*\boldsymbol{\Phi}^\top$, and $\hat{\boldsymbol{c}}$ by $\hat{\boldsymbol{\tau}}^c$, respectively.

We will now show how we can use the learned statistical model to build different active learning modalities.

5.4 Active learning modalities

In this section, we derive two active learning strategies from the learned joint model: imitation and intrinsically-motivated learning, and a criterion for choosing which learning modality is better suited at the current learning stage. To facilitate the presentation of the approach, we will introduce the approach in the context of a specific robot experiment, where the aim is to learn to move an object to different positions. First, we present the task and the goal of the active learning framework. Secondly, we present the proposed method for active imitation learning. Then, we propose a method for active intrinsically-motivated learning. Finally, a criterion for actively choosing whether imitation or intrinsically-motivated learning is better suited is presented.

5.4.1 Manipulation task

We present our approach in the context of learning to manipulate an object with a robot. The trajectory is composed of the robot joint states $\boldsymbol{\tau}^{\text{robot}}$ and the object position $\boldsymbol{\tau}^{\text{obj}}$, which implies that the ProMP weights $\boldsymbol{w}$ are a concatenation of robot weights $\boldsymbol{w}^{\text{robot}}$ and object weights $\boldsymbol{w}^{\text{obj}}$.

The goal of the task is to move the object to different desired final object positions $\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}$. We denote the goal space $\mathcal{G}$ as the space of all desired final object positions we would like our robot to be able to generalize to. Formally, this means that there exists an unknown ground truth target distribution $p^{\text{GT}}(\boldsymbol{w}) = p^{\text{GT}}(\boldsymbol{w}^{\text{robot}}, \boldsymbol{w}^{\text{obj}})$ which can be used to generate robot movements $p^{\text{GT}}(\boldsymbol{w}^{\text{robot}}|\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T})$ that bring the object to the position $\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}$.

We aim to learn this unknown joint distribution by combining imitation and intrinsically-motivated learning.

5.4.2 Imitation learning

We suppose here that there exists a human demonstrator/oracle that can be queried to demonstrate a robot movement that brings the object to any desired final position $\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}$ in $\mathcal{G}$. Acquiring these demonstrations is usually cumbersome, therefore we would like the demonstrations to be as informative as possible. We propose to choose the demonstration with active learning to alleviate

this limitation.

Given a current database of movements $\tilde{\boldsymbol{W}}$, we propose to leverage the uncertainties learned by the BGMM and choose the goal $\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}$ for which the entropy of the epistemic part of the conditional distribution $p(\boldsymbol{w}^{\text{robot}}|\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})$ is maximal. As explained in the previous section, this entropy is not easy to compute for GMMs, so we instead maximize a closed-form lower bound. The full active imitation learning algorithm is shown in Algorithm 6. Note that the process is very similar to what was proposed in previous chapter, the only difference here is that the context is internal (final object position) and not external.

Algorithm 6: Active imitation learning

Data: Movement database $\tilde{\boldsymbol{W}} = \{\boldsymbol{w}_i^{\text{robot}}, \boldsymbol{w}_i^{\text{obj}}\}_{i=1}^N$, goal space $\mathcal{G}$
Result: goal $\boldsymbol{\tau}_{\text{des}^*}^{\text{obj},t=T}$ at which to request a demonstration

Learn joint distribution of $p(\boldsymbol{w}|\tilde{\boldsymbol{W}}) = p(\boldsymbol{w}^{\text{robot}}, \boldsymbol{w}^{\text{obj}}|\tilde{\boldsymbol{W}})$ with BGMM;
Calculate $p(\boldsymbol{w}^{\text{robot}}|\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})$ using Eqs (4.4) to (4.8);
Isolate the epistemic uncertainty $p^{ep}(\boldsymbol{w}^{\text{robot}}|\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})$ with Eqs (4.9) and (4.11);
Approximate the entropy of $p^{ep}(\boldsymbol{w}^{\text{robot}}|\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})$ with Eqs (4.12) to (4.14);

Find $\boldsymbol{\tau}_{\text{des}^*}^{\text{obj},t=T} = \arg\max_{\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T} \in \mathcal{G}} \left[H_{lower}\big(p^{ep}(\boldsymbol{w}^{\text{robot}}|\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})\big)\right]$.

5.4.3 Intrinsically-motivated learning

We present here another learning modality, where the robot can try out a movement by itself and observe the environment changes in an open-ended manner. Namely, the robot chooses to execute a particular movement and observes the movement of the object. In contrast to imitation learning, one major advantage of intrinsically-motivated learning is that it does not require the presence of a human demonstrator.

We propose to select a robot movement based on how uncertain we are about the object movements it will cause. Formally, we would like to try the robot movement that maximizes the entropy of the epistemic part of the conditional distribution $p(\boldsymbol{w}^{\text{obj}}|\boldsymbol{w}^{\text{robot}}, \tilde{\boldsymbol{W}})$, but this poses several problems. From a robotics point of view, doing so might pose safety problems as the movement retrieved might be very far from the underlying distribution $p^{\text{GT}}(\boldsymbol{w}^{\text{robot}})$ we aim to learn. From an active learning point of view, our active learning selection scheme is myopic and such criterion might select robot movements far away from the underlying distribution, i.e., where no generalization is required. For these reasons, we propose to use an information-density method [15]. Namely, we aim to find a robot movement that both has high information content (in the sense of the epistemic entropy), and that is close to the distribution of robot movements $p^{\text{robot}}(\boldsymbol{w}^{\text{robot}}|\tilde{\boldsymbol{W}})$:

$$\boldsymbol{w}^{\text{robot}^*} = \arg\max_{\boldsymbol{w}^{\text{robot}} \in \mathcal{W}^{\text{robot}}} \Big[H_{lower}\Big(p^{ep}(\boldsymbol{w}^{\text{obj}}|\boldsymbol{w}^{\text{robot}}, \tilde{\boldsymbol{W}})\Big) + \beta p^{\text{robot}}(\boldsymbol{w}^{\text{robot}})\Big], \tag{5.1}$$

where β is an hyperparameter weighting the relative importance of the two costs.

The full intrinsically-motivated learning algorithm is shown in Algorithm 7.

Algorithm 7: Active intrinsically-motivated learning

Data: Movement database $\tilde{\boldsymbol{W}} = \{\boldsymbol{w}_i^{\text{robot}}, \boldsymbol{w}_i^{\text{obj}}\}_{i=1}^N$, robot movement space $\mathcal{W}^{\text{robot}}$
Result: robot movement $\boldsymbol{w}^{\text{robot}^*}$ to execute

Learn joint distribution of $p(\boldsymbol{w}|\tilde{\boldsymbol{W}}) = p(\boldsymbol{w}^{\text{robot}}, \boldsymbol{w}^{\text{obj}}|\tilde{\boldsymbol{W}})$ with BGMM;
Calculate $p(\boldsymbol{w}^{\text{obj}}|\boldsymbol{w}^{\text{robot}}, \tilde{\boldsymbol{W}})$ using Eqs (4.4) to (4.8);
Isolate the epistemic uncertainty $p^{ep}(\boldsymbol{w}^{\text{obj}}|\boldsymbol{w}^{\text{robot}}, \tilde{\boldsymbol{W}})$ with Eqs (4.9) and (4.11);
Approximate the entropy of $p^{ep}(\boldsymbol{w}^{\text{obj}}|\boldsymbol{w}^{\text{robot}}, \tilde{\boldsymbol{W}})$ with Eqs (4.12) to (4.14);
Get the marginal distribution $p^{\text{robot}}(\boldsymbol{w}^{\text{robot}}|\tilde{\boldsymbol{W}})$ from $p(\boldsymbol{w}|\tilde{\boldsymbol{W}})$;

Find $\boldsymbol{w}^{\text{robot}^*} = \arg\max_{\boldsymbol{w}^{\text{robot}} \in \mathcal{W}^{\text{robot}}} [H_{lower}(p^{ep}(\boldsymbol{w}^{\text{obj}}|\boldsymbol{w}^{\text{robot}}, \tilde{\boldsymbol{W}})) + \beta p^{\text{robot}}(\boldsymbol{w}^{\text{robot}})]$.

5.4.4 Choosing the learning modality

We have presented two different learning modalities: imitation learning and intrinsically-motivated learning[I]. We propose here a method to choose between these learning modalities.

A difficulty in choosing the right learning modality is that the epistemic entropies are not comparable for the two learning modalities. Indeed, for imitation learning we focus on the epistemic entropy of the robot movement conditional distribution for a given object final position, whereas for intrinsically-motivated learning we look at the epistemic entropy of the object movement conditional distribution for a given robot movement.

We propose to compare these learning modalities in terms of the expected reduction of the epistemic entropies of the robot movement given the desired goal. This means that we aim to minimize the expected (over the goal space) epistemic entropy on the robot movement when conditioning on the desired goal. This notion of expected epistemic entropy corresponds to

$$EE\Big(\tilde{\boldsymbol{W}}\Big) = \mathbb{E}_{\boldsymbol{\tau}_{\text{des}}^{\text{obj}, \boldsymbol{t}=\boldsymbol{T}} \in \mathcal{G}} \Big[H_{lower}\Big(p^{ep}(\boldsymbol{w}^{\text{robot}}|\boldsymbol{\tau}_{\text{des}}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})\Big)\Big]. \tag{5.2}$$

This expected epistemic entropy permits us to introduce the notion of expected epistemic entropy reduction, which is the reduction of the expected epistemic entropy when adding a datapoint $\boldsymbol{w}_{\text{new}}$ to the database $\tilde{\boldsymbol{W}}$:

$$EER\Big(\boldsymbol{w}_{\text{new}}\Big|\tilde{\boldsymbol{W}})\Big) = EE\Big(\tilde{\boldsymbol{W}}\Big) - EE\Big(\tilde{\boldsymbol{W}} \cup \{\boldsymbol{w}_{\text{new}}\}\Big). \tag{5.3}$$

[I]Note that both modalities are based on the same joint model of the movements that has been learned using a BGMM. What changes in those scenarios is the input on which we condition, which can be the desired final object position or the robot movement.

In practice, computing the expected epistemic entropy reduction involves the relearning of the BGMM with the augmented dataset $\tilde{\boldsymbol{W}} \cup \{\boldsymbol{w}_{\text{new}}\}$ and computing the expected epistemic entropy on this new joint model. This notion of expected epistemic entropy can straightforwardly be extended to a distribution[II] of potential new datapoints $p_{\text{new}}(\boldsymbol{w})$ with

$$EER\Big(p_{\text{new}}(\boldsymbol{w})\Big|\tilde{\boldsymbol{W}}\Big) = \mathbb{E}_{\boldsymbol{w}_{\text{new}}\sim p_{\text{new}}} EER\Big(\boldsymbol{w}_{\text{new}}\Big|\tilde{\boldsymbol{W}}\Big). \tag{5.4}$$

We will show now how we can use this to calculate the expected reduction of epistemic entropy when choosing imitation learning or intrinsically-motivated learning.

Imitation learning Algorithm 6 returns the goal $\boldsymbol{\tau}_{\text{des}^*}^{\text{obj},t=T}$ that should yield the most informative demonstration. Even though we do not know in advance what demonstration $\boldsymbol{w}_{\text{new}}$ we will get when querying the demonstrator, we can use our model to compute the distribution of potential demonstrations $p(\boldsymbol{w}_{\text{new}}|\boldsymbol{\tau}_{\text{des}^*}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})$ bringing the object to the desired goal. This allows us to compute the expected epistemic entropy reduction if choosing the imitation learning strategy with

$$EER(\text{Imitation}) = EER\Big(p(\boldsymbol{w}_{\text{new}}|\boldsymbol{\tau}_{\text{des}^*}^{\text{obj},t=T}, \tilde{\boldsymbol{W}})\Big|\tilde{\boldsymbol{W}}\Big). \tag{5.5}$$

Intrinsically-motivated learning Similarly, Algorithm 7 returns the robot movement $\boldsymbol{w}^{\text{robot}^*}$ expected to show an interesting object movement. We can also estimate the expected trajectories $p(\boldsymbol{w}_{\text{new}}|\boldsymbol{w}^{\text{robot}^*}, \tilde{\boldsymbol{W}})$ when executing this robot movement. From this distribution, we compute the expected epistemic entropy reduction if choosing intrinsically-motivated learning with

$$EER(\text{Intrinsic}) = EER\Big(p(\boldsymbol{w}_{\text{new}}|\boldsymbol{w}^{\text{robot}^*}, \tilde{\boldsymbol{W}})\Big|\tilde{\boldsymbol{W}}\Big). \tag{5.6}$$

In the above, we have proposed a measure to quantify the informativeness of the different learning strategies, which we can use to choose the most appropriate strategy by selecting the one which leads the highest expected epistemic entropy reduction. The selection process of the best learning strategy is summarized in Algorithm 8.

5.5 Experiments

In this section, we show the usefulness of our approaches in the context of a robotic task. First, we present the waste throwing task we consider. Then, we evaluate quantitatively the performance of our approaches for imitation learning, intrinsically-motivated learning, and the combination of both.

[II] In practice for computational reasons, we approximate $EER\Big(p_{\text{new}}(\boldsymbol{w})\Big|\tilde{\boldsymbol{W}}\Big)$ by $EER\Big(\boldsymbol{w}_{\text{new}}^{\text{MP}}\Big|\tilde{\boldsymbol{W}})\Big)$, where $\boldsymbol{w}_{\text{new}}^{\text{MP}}$ denotes the most probable datapoint under $p_{\text{new}}(\boldsymbol{w})$.

Algorithm 8: Choice of learning strategy

Data: Movement database $\tilde{\boldsymbol{W}} = \{\boldsymbol{w}_i^{\text{robot}}, \boldsymbol{w}_i^{\text{obj}}\}_{i=1}^{N}$, goal space $\mathcal{G}$, robot movement space $\mathcal{W}^{\text{robot}}$

Result: the learning strategy (*Imitation* or *Intrinsically-motivated*) that is better suited

Find $\boldsymbol{\tau}_{\text{des}^*}^{\text{obj},t=T}$ with Alg.6;
Compute the expected epistemic uncertainty reduction of imitation learning $EER(\text{Imitation})$ with Eq.5.5;

Find $\boldsymbol{w}^{\text{robot}^*}$ with Alg.7;
Compute the expected epistemic uncertainty reduction of intrinsically-motivated learning $EER(\text{Intrinsic})$ with Eq.5.6;

if $EER(\textit{Imitation}) > EER(\textit{Intrinsic})$ **then**
  Return *Imitation*
else
  Return *Intrinsically-motivated*
end

5.5.1 Waste throwing task

We consider the task of throwing waste with a 7 DoF Franka Emika Panda robot simulated in pyBullet [30]. This task is essential for the broader challenge of automatizing various forms of recycling. It is also relevant in diverse industrial applications requiring a robot to sort objects fast within a limited workspace.

An overview of the simulated setup can be seen in Fig. 5.1. The goal of the task is to be able to generate robot movements that bring a simulated can to different desired positions within a goal space $\mathcal{G}$. The particularity of this goal space is that, for a part of it, it is possible to bring the object with a non-dynamic movement because the desired final position is in the reachable robot workspace. However, for the rest of the goal space, the final desired object position is outside of the robot workspace, so that it requires the robot to throw the can with a dynamic movement. For benchmarking and reproducibility purposes, we build our experiments on a precomputed database of demonstrations. We create 200 non-dynamic demonstrations and 260 dynamic demonstrations using an oracle, that we gather in a database of demonstrations $\mathcal{D}$. In Fig. 5.1, we illustrate the can trajectory for three dynamic demonstrations and three non-dynamic demonstrations. In Fig. 5.2, we show the final can positions in our database, with the blue color representing the non-dynamic demonstrations and the orange color representing the dynamic demonstrations.

The trajectories of our database encode the robot movement at a frequency of 240Hz, with $T = 639$ timesteps, representing movements of about 3 seconds. We choose a 10-dimensional state space containing the 7 joint angle values of the robot, and the 3-dimensional Cartesian

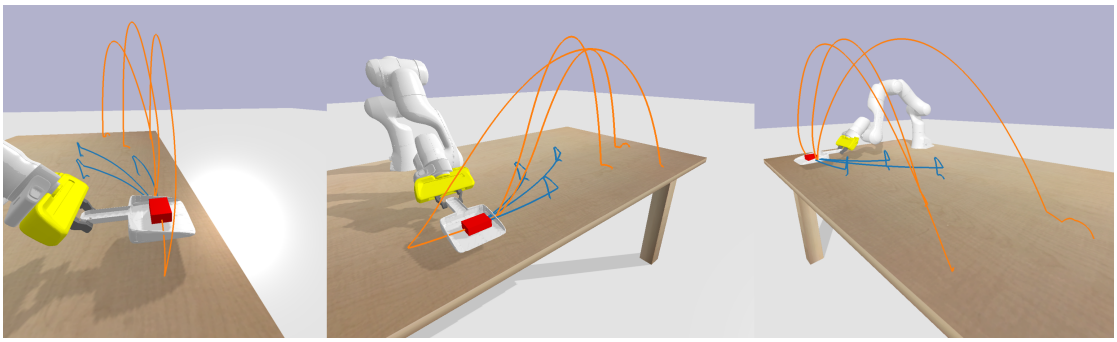


Figure 5.1 – Object trajectory for 6 demonstrations of the database (3 dynamic demonstrations in orange, and 3 non-dynamic demonstrations in blue).

position of the can. In all experiments, we use $N = 30$ Gaussian radial basis functions[III] (RBFs) for ProMP. The width of the RBFs are set as $h = (\frac{T-1}{N})^2$, and the centers $\{c_m\}_{m=1}^{D}$ are evenly spaced between $-2h$ and $T + 2h$. We choose a diagonal covariance matrix prior, with a standard deviation of 0.1 for the ProMP weights, and a mean concentration prior of 0.0001. We use a maximum number of 5 Gaussians, or strictly less than the number of demonstrations if there are less than 6 demonstrations. Other hyperparameters of the BGMM are the default hyperparameters of the *scikit-learn* library [109].

The maximization procedure in active imitation learning and active intrinsically-motivated learning is performed using a Bayesian optimization algorithm: the Tree-Structured Parzen Estimator approach (TPE) [10], implemented in the Python package hyperopt [11]. A maximal number of iterations of 100 is used in the algorithm. For imitation learning, we use a 2-dimensional uniform search space corresponding to the goal space. For intrinsically-motivated learning, as the space of possible robot movements is of high dimension (30 basis functions $\times$ 7 joint angles), we perform the search on the first two principal components of $\{\boldsymbol{w}_i^{\text{robot}}\}_{i=1}^{N}$, found by principal component analysis (PCA) [143]. The search space that we use is then the marginal distribution $p(\boldsymbol{w}^{\text{robot}})$ projected to the 2-dimensional PCA subspace.

We introduce an objective metric for comparing our learning modalities: the task cost, which is simply a ℓ_2 norm between the final object position and the desired object position, averaged over the goal space. In practice, we compute this task cost by computing the maximum *a posteriori* robot movement given a goal chosen over a uniform grid of 5×5 goals in the goal space, execute those 25 movements in simulation, and average the ℓ_2 norms between the final object positions and the desired object positions. Such a metric presents the advantage of being directly representative of the quality of the learned task, while remaining agnostic to the metrics we chose for active learning. It is important to note here that this metric based on an external reward is used only for comparison, and not by our active learning algorithms.

[III] Namely: $\Phi_m(t) = \frac{\phi_m(t)}{\sum_{n=1}^{D} \phi_n(t)}$ with $\phi_m(t) = \exp\big(-\frac{(t-c_m)^2}{2h}\big)$.

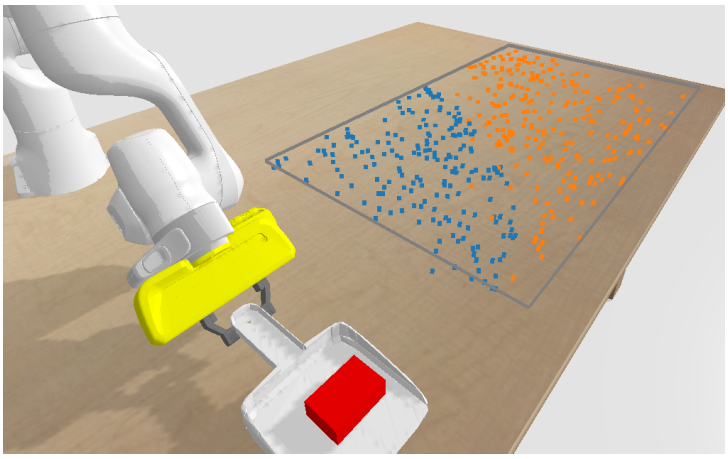


Figure 5.2 – Desired final object positions. The grey rectangle represents the goal space $\mathcal{G}$. Blue/orange dots show the final object position of respectively the non-dynamic/dynamic demonstrations of the database.

5.5.2 Imitation learning

We present here the results of our method in an imitation learning scenario.

First, we show qualitatively in Fig. 5.3 our method during 20 iterations of active learning, starting with 2 random initial demonstrations. We can see in this figure that our method effectively selects goals that are far from goals already observed in available demonstrations. Now, we propose to evaluate our method quantitatively. We benchmark our method against two different active learning baselines:

- Random: this baseline simply selects a random goal g from $\mathcal{G}$.
- Minimum likelihood (Min. Lik.): this method, similar to [29], chooses the goal that is the furthest from our current task representation. Formally, this means that we compute the marginal distribution of our BGMM over the goal space, and choose the goal that has the minimum likelihood under this distribution.

We initialize the learning process with 2 initial demonstrations randomly sampled from the database. For our method and the baselines, the experiment is reproduced 20 times, starting from different initial demonstrations. The results are shown in Fig. 5.4. We can see that our method outperforms both baselines in terms of task cost reduction across the learning process. Notably, it performs around 30% better than the random strategy at all stages of the learning process (at 5, 10, 15, and 20 iterations), and about 50% better than the minimum likelihood strategy. This shows that the epistemic uncertainty seems to be a good criterion for goal selection. Also, it confirms the usefulness of this low-level arbitration capability deciding where the agent currently needs to request a demonstration.

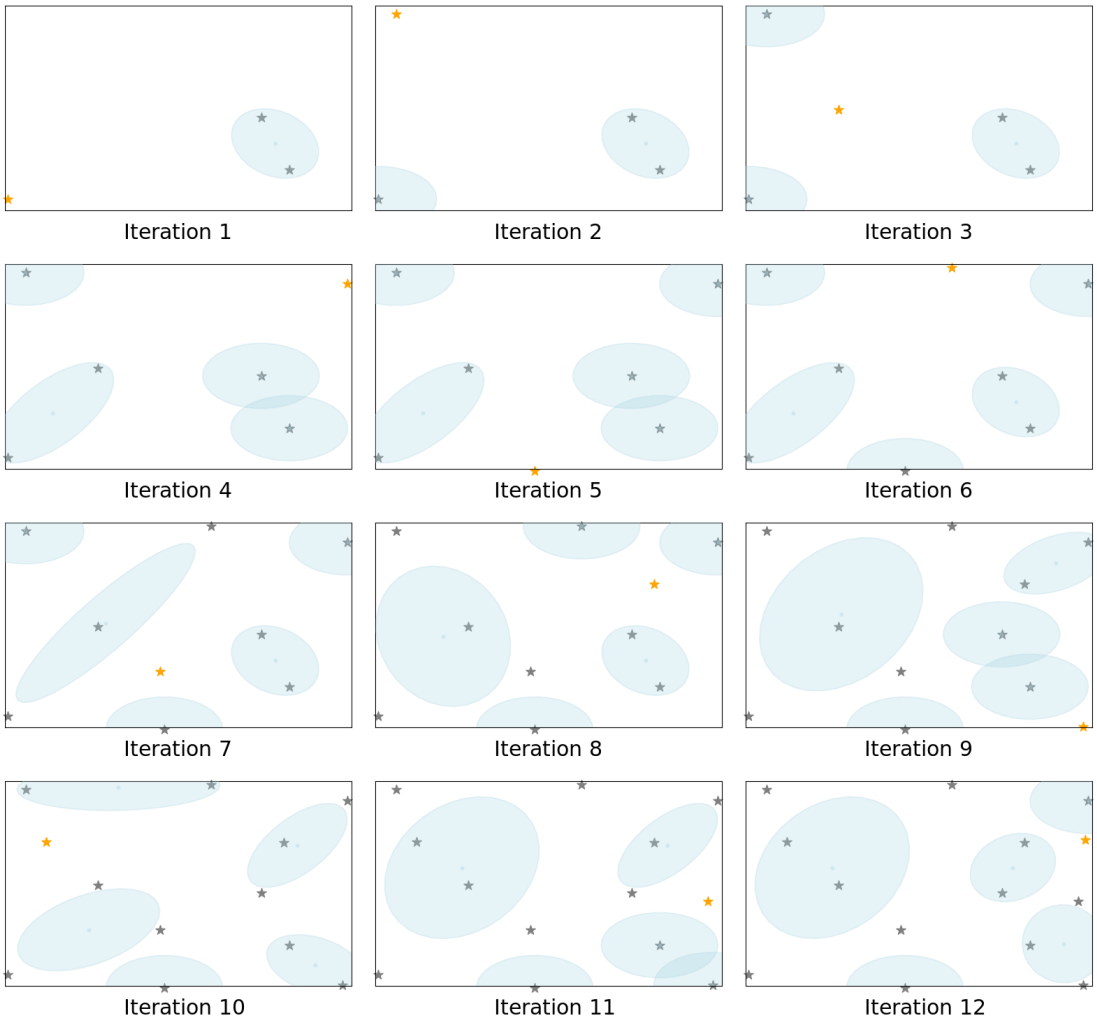


Figure 5.3 – Evolution of the active imitation learning strategy. The goal space is represented in this figure. Grey stars represent the final object position of the available demonstrations, and orange stars the selected goal to query. The transparent ellipses show the marginal distribution of the BGMM on the goal space.

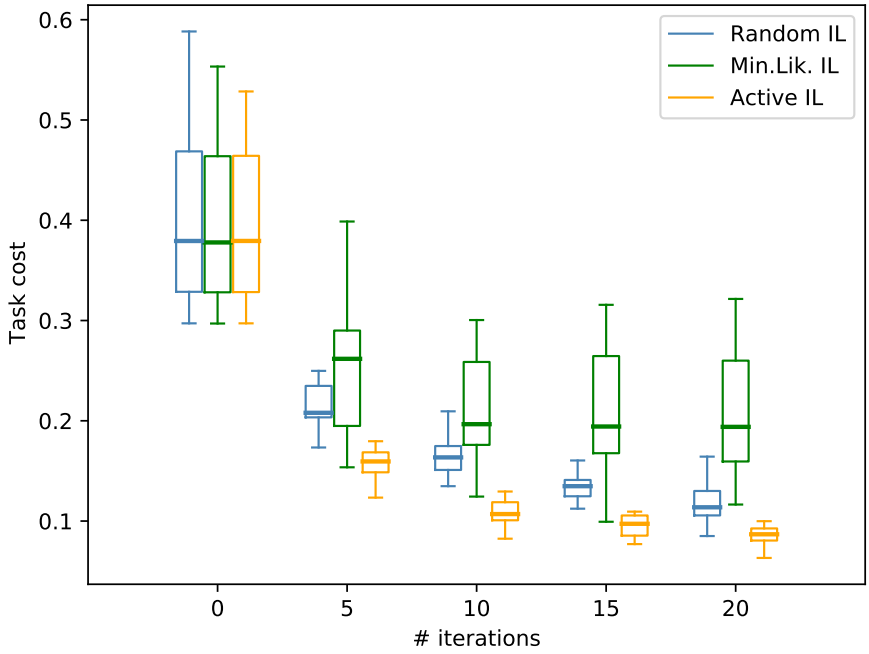


Figure 5.4 – Evaluation of imitation learning strategy.

5.5.3 Intrinsically-motivated learning

We present here the results of our intrinsically-motivated learning method. First, we would like to emphasize quantitatively the need for combining imitation learning and intrinsically-motivated learning for this waste throwing task. Namely, we want to show that using intrinsically-motivated learning can effectively reduce the task cost. We show in Fig. 5.5 the task cost (averaged over 20 demonstrations) for:

- 10 random demonstrations;
- 10 random demonstrations + 20 active intrinsically-motivated trials;
- 30 random demonstrations.

We can see that, starting from 10 initial demonstrations, 20 intrinsically-motivated learning trials can improve the model. We can notably see that 20 intrinsically-motivated trials reduce the task cost half as well as 20 additional demonstrations. This shows that intrinsically-motivated learning can be used to reduce the burden of the human demonstrator by reducing the number of demonstrations s/he will be asked. Namely, Fig. 5.5 shows that intrinsically-motivated learning seems to be a good learning modality to be combined with imitation learning. Also, note that intrinsically-motivated trials are less informative than demonstrations, which is intuitive since an intrinsically-motivated trial explores locally around the demonstrations, and hence is less informative than demonstrations in unknown areas. We propose now a baseline to compare our

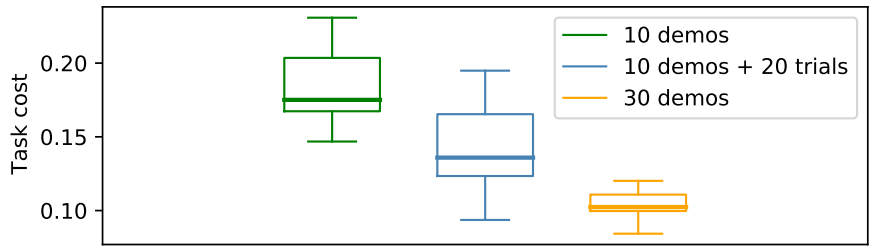


Figure 5.5 – Influence of demonstrations for intrinsically-motivated learning strategy.

intrinsically-motivated learning method with:

- Random: This baseline computes the marginal $p(\boldsymbol{w}_{\text{robot}}|\tilde{\boldsymbol{W}})$ from the BGMM, and samples a robot movement from it. This seems like a reasonable baseline which already uses the correlations in the observed robot movements, and samples meaningful robot movements that are close to the observed demonstrations.

In Fig. 5.6, we show the performance of our method compared to this baseline, averaged over 20 experiments, and starting from 5 or 10 randomly sampled initial demonstrations. We can observe that our method presents a clear improvement over the baseline in both cases. Namely, the baseline deteriorates the task cost across the iterations, whereas our method permits to reduce the task cost, as observed in Fig. 5.5 (the mean task cost is reduced by around 20% after 10 autonomous trials in both cases). The deterioration of the task cost with the random approach can be explained by the fact that sampling from the marginal distribution of the robot movements at each iteration might end up with samples that are quite far from the original distribution, hence not useful for the task.

5.5.4 Choice of learning modality

Here, we show the usefulness of choosing actively the learning modality at each iteration of the learning process. Our results, averaged over 20 experiments, start with 2 initial demonstrations (randomly sampled). In Fig. 5.7, we show which learning modalities are chosen by our method during the learning process. We can see that, for the first 5 iterations, the imitation learning strategy is almost always preferred, while afterwards the two learning modalities are selected with about the same probability. On average, the intrinsically-motivated learning modality is chosen with a probability of 36%. Leveraging this knowledge, we introduce a baseline which simply chooses the intrinsically-motivated learning strategy in a random manner with a probability 0.36, and imitation learning otherwise. Note that this baseline is already quite good, as it involves the information of the optimal probability of selecting the intrinsically-motivated strategy obtained with our method. The results are shown in Fig. 5.8. We observe that our method outperforms this baseline in the beginning of the learning process (at iteration 5), but gives similar results later in the training process. This suggests that our method for choosing the learning modality is

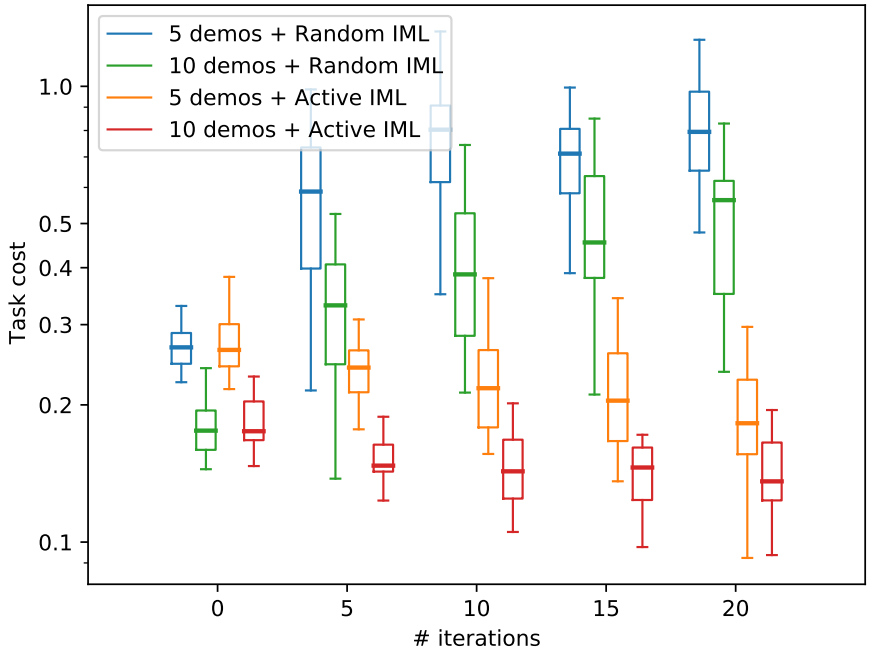


Figure 5.6 – Evaluation of intrinsically-motivated learning strategy (task cost in logarithmic scale).

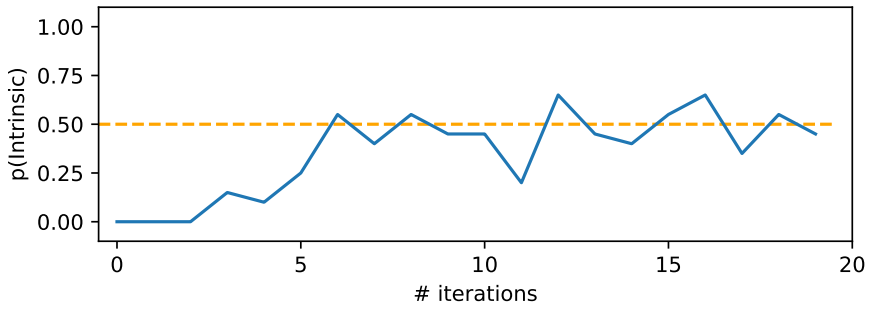


Figure 5.7 – Example of a learning process in which the learning strategy is selected at each step based on the proposed active learning method.

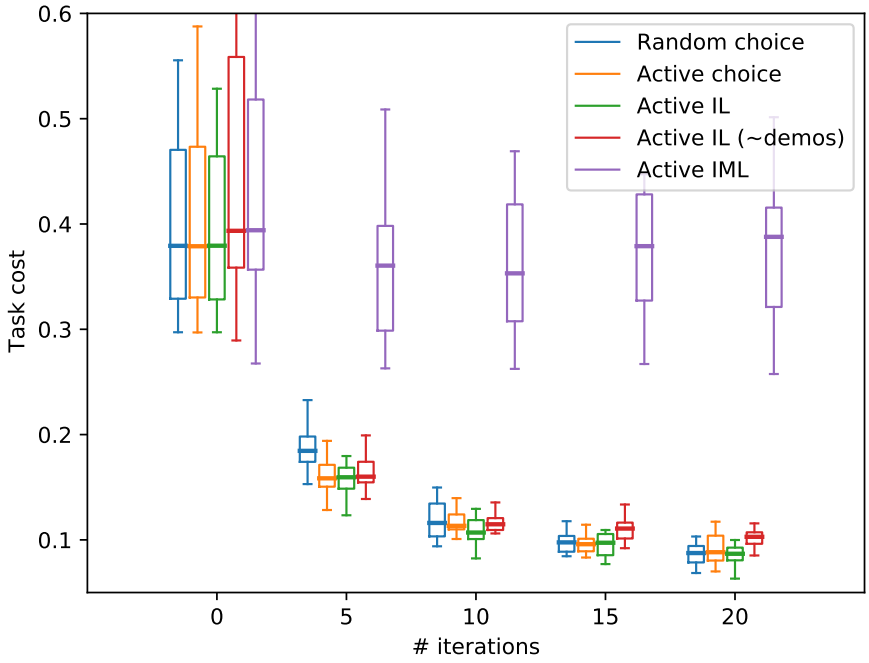


Figure 5.8 – Evaluation for the choice of the learning strategy.
(**Blue**: Random choice between Imitation Learning (IL) and Intrinsically-motivated Learning (IML), with probability 36% of choosing IML. **Orange**: Active choice of the learning modality. **Green**: Selecting always imitation learning. **Red**: Selecting always imitation learning, but using the same number of demonstrations as the active arbitration strategy (Orange boxplot). **Purple**: Selecting always intrinsically-motivated learning modality.)

useful for the investigated task, especially in the beginning of the learning process. In Fig. 5.8, we also show the performance of two additional baselines choosing always the same learning modality. We can see that choosing always *intrinsically-motivated learning* results in very poor learning. This is because two initial demonstrations are not sufficient to be able to generate meaningful movement variations. This is consistent with the fact that *imitation learning* should be preferred in the beginning of the learning process, as our method has automatically discovered (see Fig. 5.7). We also observe in Fig. 5.8 that choosing actively the learning modalities results in a task cost on par with only *imitation learning* across the whole learning process, which is a nice result because it means that we can reduce the number of demonstrations by 36% without suffering from a performance degradation, and therefore reduce the human burden of providing demonstrations. A fairer comparison is to compare our method against only *imitation learning* with the same number of demonstrations [IV], which we also plotted in Fig. 5.8. We can see that our method outperforms this baseline at iterations 15 and 20 by around 15%. This therefore motivates the meta-level arbitration capability of our framework for orchestrating the different learning modalities.

[IV] Namely 0, 5, 8, 10, 13 demonstrations at iterations 0, 5, 10, 15, 20.

5.6 Conclusion

In this chapter, we used the Bayesian representation of robot movements extending the framework of probabilistic movement primitives that we presented in previous chapter. With this Bayesian representation, we proposed three active learning criteria leveraging the knowledge of the model uncertainties (epistemic uncertainties) that permit two different learning modalities (imitation learning and intrinsically-motivated learning) as well a principled method for arbitrating between them in an open-ended manner. To the best of our knowledge, our work is the first to integrate those three aspects.

We showed the robustness of our approach with a waste throwing task with a 7-DoF simulated Franka Emika Panda robot. We studied the usefulness of each of our active learning algorithms by comparing them to alternative baselines, and showed that in all experiments, our algorithms give the best performance.

The fundamental element of our method lies in that we model the joint distribution of the movement. By doing so, we can compute several forms of conditional distributions (in our case, quantifying the effect of a specific robot movement on the object for intrinsically-motivated learning, or the robot movement needed to bring the object to a desired final position for imitation learning). Also, as intrinsically-motivated learning and imitation learning are based on the same joint model of the movement, we have shown that we can compare these very different learning modalities quantitatively.

6 Summary and future work

Learning from Demonstration has emerged as a promising framework for robotics democratization, enabling non-expert users to easily (re)program robots. In this thesis, we have proposed learning methods addressing some of the open questions currently limiting the potential of LfD applications.

First, we have focused on the usual need for users to align demonstrations and to appropriately choose the type of basis functions. By relying on Fourier series, which can approximate any signal, the user does not have to care anymore about providing an adequate set of basis functions, rich enough to represent the demonstrations, and small enough to permit statistics to be performed efficiently. Such task is indeed non-trivial for users, even if they are expert. Also, such framework removes the need to align demonstrations, which usually has to be done either by the user or by a separate algorithm that may introduce other sources of errors and whose result should be checked by the user. We believe that this is therefore a useful framework that takes a step towards LfD's ultimate goal: enabling non-expert users to program robots. We have successfully demonstrated the usefulness and applicability of this method on rhythmic robotic tasks involving complex patterns.

Typical LfD frameworks leverage the variability observed in the demonstrations to be able to adapt/generalize to new situations that were not demonstrated. It is therefore crucial to provide demonstrations that present a wide variety of variations of the task to be learned. It is, however, not trivial to quantify what constitutes a good demonstration from the point of view of the robot, especially for non-expert users with no knowledge about the underlying LfD algorithms. That's why we proposed a robot-centric active learning method in Chapter 4. Our method extends the framework of probabilistic movement primitives with a Bayesian view. Such Bayesian representation permits to quantify what constitutes a useful demonstration in terms of the current uncertainties of the model about it. We have demonstrated the superiority of this approach with respect to the state of the art, and its usefulness both in simulation and on a real robot. We have shown that our active learning method permits to achieve better generalization capabilities for a given number of demonstrations. We therefore believe that such a framework could permit to

learn more complicated tasks than a standard passive LfD framework.

By choosing actively the demonstrations to show, more complicated skills can be learned. But the number of demonstrations a user is willing to give is low (researchers typically target below 20), which can make some tasks that, e.g., require a lot of precision, or involve very diverse movements to perform depending on the situation, difficult to learn purely from demonstrations. In the same way that humans do not learn from a single learning modality, we have shown that it is beneficial to combine learning from demonstrations with other learning modalities. We have proposed in third chapter an active learning method involving several learning modalities: learning from demonstrations, intrinsically-motivated learning, as well as their arbitration. We have built upon the Bayesian movement representation proposed in Chapter 4 and proposed two new active learning schemes relying on the uncertainties captured by the Bayesian model: an active intrinsically-motivated learning criterion and a way to actively choose between imitation learning and intrinsically-motivated learning. We have shown the applicability of our approach on a complex simulated waste throwing task that involves two different types of movements (non-dynamic motion when the final location is in the robot workspace, dynamic motion otherwise).

As a conclusion, we have proposed in this thesis methods to learn representations and strategies that reduce the workload of users when programming robot by demonstrations.

6.1 Possible research directions

We now discuss the potential research directions that could be considered for future work.

6.1.1 Fourier movement primitives for discrete motions

Our method based on Fourier decomposition presented in Chapter 3 has only been tested on rhythmic movements so far. We believe that, due to the theoretical properties of the Fourier decomposition, it could be also be interesting for discrete (point-to-point) motions. Typical approaches need to align such demonstrations using a separate algorithm such as Dynamic Time Warping [92]. Indeed, performing statistics on non-aligned demonstrations might result in very poor performance. A basic way to use Fourier movement primitives for discrete motions could follow the following steps:

1. Make sure demonstrations have the same length T (linearly interpolate them if not)
2. Symmetrize the demonstrations to signals of length $2T$ (where the first half is the demonstration and the second the time-reversed demonstration)
3. Perform statistics on the Fourier decomposition of those periodic signals

We believe that such approach could better deal with misalignments than standard basis functions. Though our approach proposed in Chapter 3 can easily be extended to discrete motions, the

usefulness, generalizability and adaptability to this new use case remains to be demonstrated. Also, the method we proposed for making statistics in the Fourier domain assumes that the signals considered have the same length. Several ways to transform demonstrations of different lengths to periodic signals of the same length could be considered, and their relevance for the goal of making statistics would have to be studied. Also, note that the method we proposed for adapting to new situations would probably not be suited for discrete movements. Standard conditioning would probably perform better in this case, so there would be a need to overcome the numerical problems that arose when conditioning on the high-dimensional Fourier domain, for instance using only a subset of Fourier coefficients [1], or by performing dimensionality reduction on the Fourier domain. This might also lead to better statistics in the Fourier domain.

6.1.2 Quantifying the number of demonstrations required

Our Bayesian movement representation approach proposed in Chapter 4 proposes an active learning criterion for choosing the most informative demonstration based on the epistemic uncertainties. One important question remains however open: how many demonstrations are required for a given task? We have shown in Figures 4.7, 4.8, 4.9 that the reduction of the epistemic uncertainties was indeed correlated with a reduction of the task cost, and hence with task performance. One possible way to quantify how many demonstrations are required would be to define a threshold in terms of epistemic uncertainties after which the robot stops asking for demonstrations. For instance, if the reduction of the epistemic uncertainties for the last demo(s) was too small, the learning can be stopped. While being a very simple criterion, a closer look at Figures 4.7 and 4.8 shows us that this might not work properly. Indeed, in those 2 experiments the epistemic uncertainties decreased almost linearly with the number of demonstrations, while the task cost had reached a plateau. This suggests that, even though they are correlated, epistemic uncertainties and task performance are not linearly dependent, hence the latter criterion might not be very relevant. We therefore believe that quantifying when enough demonstrations have been provided for the task to be learned is not trivial, and could be addressed in future work.

6.1.3 Considering additional learning modalities

We have focused in this thesis on imitation learning and intrinsically-motivated learning. We believe that it would be beneficial to consider other learning modalities, first at the level of the interaction with the user. Our active learning strategies maximize the usefulness of demonstrations by requesting specific ones to a user. The cognitive load of the user might be even more reduced if we were to consider other forms of interaction:

- *Partial demonstrations*

[1] Note that the Fourier basis functions provide a notion of task complexity inherently. Indeed, we could try and represent the movements using only small frequencies, and gradually increase the granularity of the movement learned by adding basis functions of higher and higher frequencies

While in our pouring and throwing experiments this might not be relevant, providing full demonstrations is cumbersome and partial demonstrations could definitely be an interesting alternative. For instance, for tasks where only the final state matters, it might be more informative to have more demonstrations of possible final states than full demonstrations. Learning from both full and partial demonstrations should be possible by adding in the optimization problem a term related to the partial demonstrations as such:

$$\theta = \arg\max_{\theta} \big(\log p(\boldsymbol{X}^{\text{full}}|\theta) + \log p(\boldsymbol{X}^{\text{partial}}|\theta) \big). \tag{6.1}$$

For solving this optimization problem, given that partial demonstrations are similar to missing data, one might need to consider the more general Hierarchical Bayesian Model learning procedure and not its simplification, as discussed in Section 2.2.

Choosing the most informative partial (e.g., final state) demonstration could be done exactly as we have proposed, by considering the marginal distribution over the time portion considered. In the same way that epistemic uncertainties could not be compared directly for imitation learning and intrinsically-motivated learning, they might not be directly comparable for full and partial demonstrations. The same approach as in Chapter 5 could be considered for choosing between those learning modalities: the one which gives the highest expected epistemic uncertainties reduction.

- *Human feedback*
  One could consider learning from human feedback instead of demonstrations, as it might be less cumbersome for a user to indicate if an executed trajectory is valid or not with a binary feedback. Learning could then be done by maximizing the likelihood/posterior probability of valid robot executions while minimizing the one of incorrect ones, similarly to what has been proposed in [56] for learning from demonstrations of what not to do. Another alternative for incorporating human feedback might be to have the robot demonstrating two trajectories, and the user choosing which one is the best. It could notably be interesting to try and adapt the framework of [102] to our Bayesian movement representation.

Another promising direction would be to consider additional robot learning modalities requiring no user, in order to have a richer set of learning modalities from which to choose:

- *Reinforcement learning*
  As discussed in Chapter 2, designing a reward function that does not lead to suboptimal behavior is not trivial. We believe that if reinforcement learning was combined with other learning modalities (e.g., learning from demonstrations), this difficulty would be greatly alleviated as the other learning modalities would provide additional guidance. Extending our Bayesian framework for dealing with this learning modality might therefore be of interest. A reinforcement signal could provide a rich guidance for the robot to autonomously explore, that might be more directed than the intrinsically-motivated reward rewarding curiosity we have proposed in Chapter 5.

A potential way to go could be to adopt a Bayesian Optimization approach [50] to reinforcement learning. Indeed, evolution strategies for robot reinforcement learning have been widely used in the literature, see [133] for some pointers.

One way to go that would exploit our Bayesian movement representation could be to model the joint distribution of the robot trajectory along with the reward $p(\boldsymbol{w}, r)$ with a BGMM. This would still be compatible with our framework, because by marginalizing out the reward this induces a joint distribution of the robot movements, as proposed in our framework. Finding the robot trajectory that would minimize the reward could then be done by choosing an acquisition function for the epistemic part of the conditional distribution $p^{\text{ep}}(r|\boldsymbol{w})$, such as the commonly used expected improvement criterion [50].

Overall, we believe that our Bayesian framework is generic and could be used for diverse learning modalities. While the most suitable learning modalities are probably problem-dependent, we believe that an interesting avenue for future work lies in designing new learning modalities, and arbitrating between them.

As the number of learning modalities will increase, it might be desirable to consider a human cost for each learning modality. In Chapter 5, our arbitration mechanism between imitation learning and intrinsically-motivated learning relied solely on the expected reduction of epistemic uncertainties. It might be relevant to weigh this with the human cost associated to each learning strategy, in order to take into account in the active arbitration mechanism that some learning modalities require more effort from the human user.

6.1.4 Model-based learning approaches

In this thesis, we have focused on model-free learning approaches. The models developed rely on the modeling of the distribution of the robot trajectories. We have shown that this could be successfully applied to diverse tasks. As discussed in Chapter 2, the choice between model-free or model-based learning approaches is problem-dependent. For the pouring experiment considered in Chapter 4, it seems clear that model-free approaches are a more parsimonious description of the task than model-based approaches that would try to model the dynamics of the fluid, which is very complex and is actually not necessary to know to execute the task. For other tasks, it might be the opposite. Let us consider for instance a simplified pushing task where an object moves along with the robot if the robot is sufficiently close, and stays still otherwise. For such task, the description of the underlying environment dynamics is pretty simple and definitely more parsimonious than a model-free approach encoding all of the movements to perform based on the initial and desired final position of the object.

Model-based approaches usually rely on a probabilistic description of the underlying models, as it provides more robustness than learning a single model. Typical approaches used in robotics consider Gaussian Processes for learning this distribution of models, which can be used for reinforcement learning [36] or imitation learning [40]. We believe that the Bayesian Gaussian

Mixture model used throughout the thesis could be an interesting alternative to GPs for dynamics learning, as it could provide the following advantages:

- Possiblity to separate aleatoric/epistemic uncertainties, which could help for avoiding regions of high noise/aleatoric uncertainty, or for designing curious behaviors by rewarding regions of high epistemic uncertainty (see for instance [130] and [124] for model-based active exploration and [28] for an illustration of why separating aleatoric and epistemic uncertaintites in model-based reinforcement learning is important).
- Potentially a better modelling of tasks with contacts. The discontinuities involved in tasks with contacts is a current challenge for model-based learning approaches. BGMMs might be better suited to represent such discontinuities than GPs that assume very smooth functions.
- Deriving controllers from the learned model might be easier because conditioning in a BGMM results in a mixture of linear systems. Such property might also be helpful for designing feedback controllers.
- The covariance prior and mean prior can be chosen appropriately to avoid unstable dynamics far from the training data. For instance, by putting the mean prior to zero one can enforce that we expect nothing to happen in the environment when far from the training data.

The latter properties remain to be tested and verified and are an interesting topic for future work. Also, methods bridging model-based and model-free principles might benefit from both worlds, and could be worth considering.

Bibliography

[1] Jacopo Aleotti and Stefano Caselli. “Robust trajectory learning and approximation for robot programming by demonstration”. In: *Robotics and Autonomous Systems* 54.5 (2006), pp. 409–413.

[2] W. Amanhoud, M. Khoramshahi, and Aude Billard. “A Dynamical System Approach to Motion and Force Generation in Contact Tasks”. In: *Robotics: Science and Systems*. Freiburg im Breisgau, Germany, June 2019.

[3] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. “A survey of robot learning from demonstration”. In: *Robotics and autonomous systems*. Vol. 57. 5. 2009, pp. 469–483.

[4] L. Armesto, J. Bosga, V. Ivan, and S. Vijayakumar. “Efficient learning of constraints and generic null space policies”. In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. May 2017, pp. 1520–1526.

[5] L. Armesto, V. Ivan, J. Moura, A. Sala, and S. Vijayakumar. “Learning Constrained Generalizable Policies by Demonstration”. In: *Robotics: Science and Systems*. Cambridge, Massachusetts, July 2017.

[6] Christopher Atkeson. “Using local models to control movement”. In: *Advances in Neural Information Processing Systems (NIPS)* 2 (1989), pp. 316–323.

[7] Hagai Attias. “A Variational Bayesian Framework for Graphical Models.” In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 12. 1999.

[8] Michael Bain and Claude Sammut. “A Framework for Behavioural Cloning.” In: *Machine Intelligence 15*. 1995, pp. 103–129.

[9] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[10] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. “Algorithms for hyper-parameter optimization”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2011, pp. 2546–2554.

[11] James Bergstra, Dan Yamins, and David 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.

[12] Daniel E Berlyne. *Conflict, arousal, and curiosity.* McGraw-Hill Book Company, 1960.

[13] A. Billard, S. Calinon, and R. Dillmann. "Handbook of Robotics". In: *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2016. Chap. Learning from Humans, pp. 1995–2014.

[14] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. "Survey: Robot programming by demonstration". In: *Springer Handbook of Robotics*. 2008, pp. 1371–1394.

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

[16] Christopher Bishop. "Variational Principal Components". In: *Proceedings of Ninth International Conference on Artificial Neural Networks (ICANN)*. 1999, pp. 509–514.

[17] Maya Cakmak, Nick DePalma, Rosa I Arriaga, and Andrea L Thomaz. "Exploiting social partners in robot learning". In: *Autonomous Robots* 29.3-4 (2010), pp. 309–329.

[18] S. Calinon and A. G. Billard. "What is the Teacher's Role in Robot Programming by Demonstration? - Toward Benchmarks for Improved Learning". In: *Interaction Studies* 8.3 (2007), pp. 441–464.

[19] S. Calinon, D. Bruno, and D. G. Caldwell. "A task-parameterized probabilistic model with minimal intervention control". In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. Hong Kong, China, May 2014, pp. 3339–3344.

[20] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard. "Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression". In: *Robotics and Automation Magazine* 17.2 (2010), pp. 44–54.

[21] Sylvain Calinon. "A tutorial on task-parameterized movement learning and retrieval". In: *Intelligent service robotics* 9.1 (2016), pp. 1–29.

[22] Sylvain Calinon. "Learning from demonstration (programming by demonstration)". In: *Encyclopedia of robotics* (2018), pp. 1–8.

[23] Sylvain Calinon. "Mixture models for the analysis, edition, and synthesis of continuous time series". In: *Mixture Models and Applications*. Springer, 2020, pp. 39–57.

[24] Sylvain Calinon. *Robot programming by demonstration*. EPFL Press, 2009.

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

[26] Caroline J Charpentier, Kiyohito Iigaya, and John P O'Doherty. "A Neuro-computational Account of Arbitration between Choice Imitation and Goal Emulation during Human Observational Learning". In: *Neuron* (2020).

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

[28] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models". In: *Advances in Neural Information Processing Systems (NIPS)*. 2018.

[29] A. Conkey and T. Hermans. "Active Learning of Probabilistic Movement Primitives". In: *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*. 2019, pp. 1–8.

[30] E. Coumans and Y. Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. http://pybullet.org. 2016.

[31] Bruno Castro Da Silva, Gianluca Baldassarre, George Konidaris, and Andrew Barto. "Learning parameterized motor skills on a humanoid robot". In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 5239–5244.

[32] Neha Das, Sarah Bechtle, Todor Davchev, Dinesh Jayaraman, Akshara Rai, and Franziska Meier. "Model-Based Inverse Reinforcement Learning from Visual Demonstrations". In: *Conference on Robot Learning (CoRL)* (2020).

[33] Edward L. Deci and Richard M. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer US, 1985.

[34] S. Degallier, L. Righetti, S. Gay, and A. J. Ijspeert. "Toward simple control for complex, autonomous robotic applications: combining discrete and rhythmic motor primitives". In: *Autonomous Robots* 31.2-3 (2011), pp. 155–181.

[35] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. "A survey on policy search for robotics". In: *Foundations and trends in Robotics* 2.1-2 (2013), pp. 388–403.

[36] Marc Deisenroth and Carl E Rasmussen. "PILCO: A model-based and data-efficient approach to policy search". In: *Proc. Intl Conf. on Machine Learning (ICML)*. 2011, pp. 465–472.

[37] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[38] Nicolas Duminy, Sao Mai Nguyen, and Dominique Duhaut. "Learning a set of interrelated tasks by using a succession of motor policies for a socially guided intrinsically motivated learner". In: *Frontiers in neurorobotics* 12 (2019), p. 87.

[39] Nicolas Duminy, Sao Mai Nguyen, Junshuai Zhu, Dominique Duhaut, and Jerome Kerdreux. "Intrinsically motivated open-ended multi-task learning using transfer learning to discover task hierarchy". In: *Applied Sciences* 11.3 (2021), p. 975.

[40] Peter Englert, Alexandros Paraschos, Marc Peter Deisenroth, and Jan Peters. "Probabilistic model-based imitation learning". In: *Adaptive Behavior* 21.5 (2013), pp. 388–403.

[41] J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal. "Encoding of periodic and their transient motions by a single dynamic movement primitive". In: *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*. Nov. 2012, pp. 57–64.

[42] 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.

[43] Marco Ewerton, Guilherme Maeda, Jan Peters, and Gerhard Neumann. “Learning motor skills from partially observed movements executed at different speeds”. In: *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 456–463.

[44] Marco Ewerton, Gerhard Neumann, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Guilherme Maeda. “Learning multiple collaborative tasks with a mixture of interaction primitives”. In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1535–1542.

[45] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. “Diversity is All You Need: Learning Skills without a Reward Function”. In: 2019.

[46] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. “Search on the replay buffer: Bridging planning and reinforcement learning”. In: *Advances in Neural Information Processing Systems (NIPS)* (2019).

[47] James Falkoff. *Democratization of Automation: The Next Generation of Industrial Robotics*. https://xconomy.com/boston/2018/01/22/democratization-of-automation-the-next-generation-of-industrial-robotics/. Accessed: 2021-08-13.

[48] Chelsea Finn, Sergey Levine, and Pieter Abbeel. “Guided cost learning: Deep inverse optimal control via policy optimization”. In: *Proc. Intl Conf. on Machine Learning (ICML)*. PMLR. 2016, pp. 49–58.

[49] Mikhail Frank, Jürgen Leitner, Marijn Stollenga, Alexander Förster, and Jürgen Schmidhuber. “Curiosity driven reinforcement learning for motion planning on humanoids”. In: *Frontiers in neurorobotics* 7 (2014), p. 25.

[50] Peter I Frazier. “A tutorial on Bayesian optimization”. In: *arXiv preprint arXiv:1807.02811* (2018).

[51] A. Gams, M. Do, A. Ude, T. Asfour, and R. Dillmann. “On-line periodic movement and force-profile learning for adaptation to new surfaces”. In: *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*. Dec. 2010, pp. 560–565.

[52] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič. “On-line learning and modulation of periodic movements with nonlinear dynamical systems”. In: *Autonomous Robots* 27.1 (2009), pp. 3–23.

[53] Alexander Gepperth and Benedikt Pfülb. “Gradient-based training of Gaussian Mixture Models in High-Dimensional Spaces”. In: *arXiv preprint arXiv:1912.09379* (2019).

[54] G. Gergely and G. Csibra. “Sylvia’s Recipe: The Role of Imitation and Pedagogy in the Transmission of Human Culture”. In: *Roots of Human Sociality: Culture, Cognition, and Human Interaction*. Ed. by N. J. Enfield and S. C. Levinson. Berg Publishers, 2006, pp. 229–255.

[55] 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.

[56] Daniel H Grollman and Aude Billard. "Donut as i do: Learning from failed demonstrations". In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. 2011, pp. 3804–3809.

[57] Jonathan Ho and Stefano Ermon. "Generative adversarial imitation learning". In: *Advances in Neural Information Processing Systems (NIPS)* 29 (2016), pp. 4565–4573.

[58] Lydia M Hopper, Emma G Flynn, Lara AN Wood, and Andrew Whiten. "Observational learning of tool use in children: Investigating cultural spread through diffusion chains and learning mechanisms through ghost displays". In: *Journal of experimental child psychology* 106.1 (2010), pp. 82–97.

[59] Victoria Horner and Andrew Whiten. "Causal knowledge and imitation/emulation switching in chimpanzees (Pan troglodytes) and children (Homo sapiens)". In: *Animal cognition* 8.3 (2005), pp. 164–181.

[60] Jon C Horvitz. "Mesolimbocortical and nigrostriatal dopamine responses to salient non-reward events". In: *Neuroscience* 96.4 (2000), pp. 651–656.

[61] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell. "Kernelized Movement Primitives". In: *International Journal of Robotics Research* 38.7 (2019), pp. 833–852.

[62] Yanlong Huang, Fares J Abu-Dakka, João Silvério, and Darwin G Caldwell. "Toward orientation learning and adaptation in cartesian space". In: *IEEE Transactions on Robotics* 37.1 (2020), pp. 82–98.

[63] Eyke Hüllermeier and Willem Waegeman. "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods". In: *Machine Learning* 110.3 (2021), pp. 457–506.

[64] A. J. Ijspeert, J. Nakanishi, and S. Schaal. "Movement imitation with nonlinear dynamical systems in humanoid robots". In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. 2002, pp. 1398–1403.

[65] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal. "Dynamical movement primitives: Learning attractor models for motor behaviors". In: *Neural Computation* 25.2 (2013), pp. 328–373.

[66] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. "Learning attractor landscapes for learning motor primitives". In: *Advances in Neural Information Processing Systems (NIPS)* (2002).

[67] Noémie Jaquier, David Ginsbourger, and Sylvain Calinon. "Learning from demonstration with model-based Gaussian process". In: *Conference on Robot Learning (CoRL)*. 2020, pp. 247–257.

[68] Claudia Jarrett. *Democratizing Automation – The Benefits of No Code Robotics*. https://www.autodesk.com/products/fusion-360/blog/how-democratized-robotics-will-change-the-way-things-are-manufactured/. Accessed: 2021-08-13.

[69] Rudolf Emil Kalman. "When is a linear control system optimal?" In: *Journal of Basic Engineering* 86.1 (1964), pp. 51–60.

[70] S. M. Khansari-Zadeh and A. Billard. "Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models". In: *Transactions on Robotics* 27.5 (2011), pp. 943–957.

[71] M. Khoramshahi and A. Billard. "A dynamical system approach to task-adaptation in physical human–robot interaction". In: *Autonomous Robots* 43.4 (2019), pp. 927–946.

[72] M. Khoramshahi, A. Laurens, T. Triquet, and A. Billard. "From Human Physical Interaction To Online Motion Adaptation Using Parameterized Dynamical Systems". In: *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 1361–1366.

[73] Jens Kober, Katharina Mülling, Oliver Krömer, Christoph H Lampert, Bernhard Schölkopf, and Jan Peters. "Movement templates for learning of hitting and batting". In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. 2010, pp. 853–858.

[74] Jens Kober and Jan Peters. "Policy search for motor primitives in robotics". In: *Learning Motor Skills*. Springer, 2014, pp. 83–117.

[75] A. Kolchinsky and B.D. Tracey. "Estimating mixture entropy with pairwise distances". In: *Entropy* 19.7 (2017), p. 361.

[76] Varun Raj Kompella, Marijn Stollenga, Matthew Luciw, and Juergen Schmidhuber. "Continual curiosity-driven skill acquisition from high-dimensional video inputs for humanoid robots". In: *Artificial Intelligence* 247 (2017), pp. 313–335.

[77] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. "Robot motor skill coordination with EM-based reinforcement learning". In: *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*. 2010, pp. 3232–3237.

[78] O.B. Kroemer, R. Detry, J. Piater, and J. Peters. "Combining active learning and reactive control for robot grasping". In: *Robotics and Autonomous systems* 58.9 (2010), pp. 1105–1116.

[79] Miguel Lázaro-Gredilla and Michalis K Titsias. "Variational heteroscedastic Gaussian process regression". In: *Proc. Intl Conf. on Machine Learning (ICML)*. 2011, pp. 841–848.

[80] Quoc V Le, Alex J Smola, and Stéphane Canu. "Heteroscedastic Gaussian process regression". In: *Proc. Intl Conf. on Machine Learning (ICML)*. 2005, pp. 489–496.

[81] Sergey Levine and Vladlen Koltun. "Guided policy search". In: *Proc. Intl Conf. on Machine Learning (ICML)*. 2013, pp. 1–9.

[82] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 421–436.

[83] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning". In: *CoRR* (2016).

[84] H. Lin, M. Howard, and S. Vijayakumar. "Learning null space projections". In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. May 2015, pp. 2613–2619.

[85] Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. "Developmental robotics: a survey". In: *Connection science* 15.4 (2003), pp. 151–190.

[86] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[87] 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.

[88] G.J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters. "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks". In: *Autonomous Robots* 41.3 (2017), pp. 593–612.

[89] James Marshall, Doug Blank, and Lisa Meeden. "An emergent framework for self-motivation in developmental robotics". In: *International Conference on Development and Learning* (2004).

[90] J.R. Medina Hernández, D. Lee, and S. Hirche. "Risk-Sensitive Optimal Feedback Control for Haptic Assistance". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2012.

[91] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.

[92] Meinard Müller. "Dynamic time warping". In: *Information retrieval for music and motion* (2007), pp. 69–84.

[93] Andrew Y Ng. *Shaping and policy search in reinforcement learning*. University of California, Berkeley, 2003.

[94] Hung Ngo, Matthew Luciw, Alexander Forster, and Juergen Schmidhuber. "Learning skills from play: artificial curiosity on a katana robot arm". In: *International joint conference on neural networks (IJCNN)*. IEEE. 2012, pp. 1–8.

[95] S. M. Nguyen and P.-Y. Oudeyer. "Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner". In: *Paladyn, Journal of Behavioral Robotics* 3.3 (2012), pp. 136–146.

[96] S.M. Nguyen, S. Ivaldi, N. Lyubova, A. Droniou, D. Gerardeaux-Viret, D. Filliat, V. Padois, O. Sigaud, P.Y. 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.

[97] Sao Mai Nguyen, Adrien Baranes, and Pierre-Yves Oudeyer. "Bootstrapping intrinsically motivated learning with human demonstrations". In: *Proc. IEEE Intl Conf. on Development and Learning (ICDL)* (2011), pp. 1–8.

[98] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, P. Abbeel, and Jan Peters. "An Algorithmic Perspective on Imitation Learning". In: *Found. Trends Robotics* 7 (2018), pp. 1–179.

[99] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. "Intrinsic motivation systems for autonomous mental development". In: *IEEE transactions on evolutionary computation* 11.2 (2007), pp. 265–286.

[100] Pierre-Yves Oudeyer and Frederic Kaplan. "What is intrinsic motivation? A typology of computational approaches". In: *Frontiers in neurorobotics* 1 (2009), p. 6.

[101] Rob Painter. *Democratization of technology has the power to change people's lives*. https://www.geospatialworld.net/blogs/democratization-of-technology-has-the-power-to-change-peoples-lives/. Accessed: 2021-08-13.

[102] Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa Sadigh. "Learning reward functions by integrating human demonstrations and preferences". In: *Proc. Robotics: Science and Systems (RSS)*. 2019.

[103] Zengxi Pan, Joseph Polden, Nathan Larkin, Stephen Van Duin, and John Norrish. "Recent progress on programming methods for industrial robots". In: *Robotics and Computer-Integrated Manufacturing* 28.2 (2012), pp. 87–94.

[104] Leo Pape, Calogero Maria Oddo, Marco Controzzi, Christian Cipriani, Alexander Förster, Maria Chiara Carrozza, and Jürgen Schmidhuber. "Learning tactile skills through curious exploration". In: *Frontiers in neurorobotics* 6 (2012), p. 6.

[105] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. "Using probabilistic movement primitives in robotics". In: *Autonomous Robots* 42.3 (2018), pp. 529–551.

[106] A. Paraschos, E. Rueckert, J. Peters, and G. Neumann. "Probabilistic movement primitives under unknown system dynamics". In: *Advanced Robotics* 32.6 (2018), pp. 297–310.

[107] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. "Probabilistic movement primitives". In: *Advances in Neural Information Processing Systems (NIPS)*. 2013, pp. 2616–2624.

[108] Alexandros Paraschos, Elmar Rueckert, Jan Peters, and Gerhard Neumann. "Model-free probabilistic movement primitives for physical interaction". In: *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 2860–2866.

[109] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[110] L. Peternel, T. Petrič, and J Babič. "Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface". In: *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*. 2015, pp. 1497–1502.

[111] L. Peternel, T. Petrič, E. Oztop, and J. Babič. "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach". In: *Autonomous Robots* 36.1-2 (2014), pp. 123–136.

[112] T. Petrič, A. Gams, A. J. Ijspeert, and L. Žlajpah. "On-line frequency adaptation and movement imitation for rhythmic robotic tasks". In: *International Journal of Robotics Research* 30.14 (2011), pp. 1775–1788.

[113] T. Petrič, A. Gams, L. Žlajpah, and A. Ude. "Online learning of task-specific dynamics for periodic tasks". In: *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*. Sept. 2014, pp. 1790–1795.

[114] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press, 2006.

[115] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. "Recent advances in robot learning from demonstration". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 297–330.

[116] L. Righetti, J. Buchli, and A. J. Ijspeert. "Dynamic hebbian learning in adaptive frequency oscillators". In: *Physica D: Nonlinear Phenomena* 216.2 (2006), pp. 269–281.

[117] K. J. Rohlfing, J. Fritsch, B. Wrede, and T. Jungmann. "How can Multimodal Cues from Child-directed Interaction Reduce Learning Complexity in Robots?" In: *Advanced Robotics* 20.10 (2006), pp. 1183–1199.

[118] Stuart Russell. "Learning agents for uncertain environments". In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 101–103.

[119] M. Salganicoff, L.H. Ungar, and R. Bajcsy. "Active learning for vision-based robot grasping". In: *Machine Learning* 23.2-3 (1996), pp. 251–278.

[120] Sam Sattel. *How Democratized Robotics will Change the Way Things are Manufactured*. https://www.autodesk.com/products/fusion-360/blog/how-democratized-robotics-will-change-the-way-things-are-manufactured/. Accessed: 2021-08-13.

[121] J. Saunders, C. L. Nehaniv, K. Dautenhahn, and A. Alissandrakis. "Self-Imitation and Environmental Scaffolding for Robot Teaching". In: *Intl Journal of Advanced Robotics Systems* 4.1 (2007), pp. 109–124.

[122] J. Schmidhuber. "Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts". In: *Connection Science* 18.2 (2006), pp. 173–187.

[123] Jürgen Schmidhuber. "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)". In: *IEEE Transactions on Autonomous Mental Development* 2.3 (2010), pp. 230–247.

[124] Matthias Schultheis, Boris Belousov, Hany Abdulsamad, and Jan Peters. "Receding horizon curiosity". In: *Conference on Robot Learning (CoRL)*. PMLR. 2020, pp. 1278–1288.

[125] A. Sena and M. Howard. "Quantifying teaching behavior in robot learning from demonstration". In: *The International Journal of Robotics Research* 39.1 (2020), pp. 54–72.

[126] 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.

[127] B. Settles. "Active learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (2012), pp. 1–114.

[128] C.E. Shannon. "A mathematical theory of communication". In: *Bell system technical journal* 27.3 (1948), pp. 379–423.

[129] A.P. Shon, D. Verma, and R.P.N Rao. "Active imitation learning". In: *Proc. AAAI Conference on Artificial Intelligence*. 2007, pp. 1–7.

[130] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. "Model-based active exploration". In: *Proc. Intl Conf. on Machine Learning (ICML)*. PMLR. 2019, pp. 5779–5788.

[131] D. Silver, J.A. 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.

[132] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (2016), pp. 484–489.

[133] Freek Stulp and Olivier Sigaud. "Robot skill learning: From reinforcement learning to evolution strategies". In: *Paladyn, Journal of Behavioral Robotics* 4.1 (2013), pp. 49–61.

[134] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[135] Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. "Safety augmented value estimation from demonstrations (SAVED): Safe deep model-based RL for sparse cost robotic tasks". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3612–3619.

[136] A. L. Thomaz and C. Breazeal. "Teachable robots: Understanding human teaching behavior to build more effective robot learners". In: *Artificial Intelligence* 172 (6-7 Apr. 2008), pp. 716–737.

[137] E. Todorov. “Optimality principles in sensorimotor control”. In: *Nature Neuroscience* 7.9 (2004), pp. 907–915.

[138] E. Todorov and M. I Jordan. “A minimal intervention principle for coordinated movement”. In: *Advances in neural information processing systems*. 2003, pp. 27–34.

[139] Aleš Ude, Andrej Gams, Tamim Asfour, and Jun Morimoto. “Task-specific generalization of discrete and periodic dynamic movement primitives”. In: *Transactions on Robotics* 26.5 (2010), pp. 800–815.

[140] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. “Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards”. In: *arXiv preprint arXiv:1707.08817* (2017).

[141] Robert W White. “Motivation reconsidered: The concept of competence”. In: *Psychological review* 66.5 (1959), p. 297.

[142] Andrew Whiten, Nicola McGuigan, Sarah Marshall-Pescini, and Lydia M Hopper. “Emulation, imitation, over-imitation and the scope of culture for child and chimpanzee”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 364.1528 (2009), pp. 2417–2428.

[143] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[144] Stephen Wright, Jorge Nocedal, et al. “Numerical optimization”. In: *Springer Science* 35.67-68 (1999), p. 7.

[145] P. Zukow-Goldring and M. A. Arbib. “Affordances, Effectivities and Assisted Imitation: Caregivers and the Directing of Attention”. In: *Neurocomputing* 70.13-15 (2007), pp. 2181–2193.

Thibaut Kulak

Experience

2017–now **PhD Thesis in Robotics/Machine Learning**, *EPFL / Idiap Research Institute*, Robot Learning and Interaction Group, supervised by Dr Sylvain Calinon, Active learning of robot skills from human demonstrations. Quantification of uncertainties in machine learning models for the combination of robot learning modalities. Applications on robot manipulation tasks.

2017 **Master Thesis**, *AI Lab of SoftBank Robotics*, 6-months Research Internship, Learning latent variables from the environment for sensorimotor prediction, i.e. making the robot understand the consequences of its actions. Unsupervised learning of the concept of position in an environment. Use of Recurrent Neural Networks and Deep Generative models. TensorFlow development

2016–2017 **Safran Morpho**, *6 months Research project*, Facial recognition, Learning a head pose estimator with both labeled images and half-labelled images (flopped images).
Deep Learning model development with Tensorflow
Domain Adaptation convolutional neural network to learn from different databases

Scientific publications

2021 **Thibaut Kulak, and Sylvain Calinon**, Combining Social and Intrinsically-Motivated Learning for Multi-Task Robot Skill Acquisition, in *IEEE Transactions on Cognitive and Developmental Systems (TCDS)*.

2021 **Thibaut Kulak, and Sylvain Calinon**, Intrinsically-Motivated Robot Learning of Bayesian Probabilistic Movement Primitives, Best Poster Award in *ICRA workshop: Towards Curious Robots: Modern Approaches for Intrinsically-Motivated Intelligent Behavior*.

2021 **Thibaut Kulak, Hakan Girgin, Jean-Marc Odobez, and Sylvain Calinon**, Active Learning of Bayesian Probabilistic Movement Primitives, in *IEEE Robotics and Automation Letters (RA-L)*.

2020 **Thibaut Kulak, João Silvério, and Sylvain Calinon**, Fourier movement primitives: an approach for learning rhythmic robot skills from demonstrations, in *Robotics: Science and Systems (RSS)*.

2018 **Thibaut Kulak, and Michael Garcia Ortiz**, Emergence of Sensory Representations Using Prediction in Partially Observable Environments, in *International Conference on Artificial Neural Networks (ICANN)*.

Education

2017–now **EPFL / Idiap Research Institute**, *Robot Learning and Interaction Group*, PhD thesis.

2014–2017 **Ecole Centrale Paris**, *Engineering school*, Data Science specialization.

2016–2017 **MVA**, *ENS*, Research master in AI.

Spring 2016 **University of Texas at Austin**, *Department of Data Science*.

2012–2014 **Intensive foundation degree**, *Mathematics*.

2009–2012 **Scientific baccalaureate**, *with distinction*.

Rue du Village 4 – 1922, Les Granges (Salvan) – Suisse
+41 786000994 • *thibaut.kulak@gmail.com* • *thibaut-kulak*

Languages

English	Fluent, TOEFL (620)
French	Native
German	C1 level

Computer skills

Python (Scikit-Learn, TensorFlow, Numpy, Django)
Matlab, Java, R, LaTeX, SQL, C++, Office Pack

Interests

Machine Learning, Deep Learning, Robotics, Computer Vision, Artificial Intelligence

Hobbies

Climbing, ski touring, cooking, guitar

Rue du Village 4 – 1922, Les Granges (Salvan) – Suisse
+41 786000994 • *thibaut.kulak@gmail.com* • *thibaut-kulak*