Probabilistic Methods for Imitation Learning in Social HRI

by

Michael Drolet

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Arts

Approved April 2023 by the
Graduate Supervisory Committee:

Heni Ben Amor, Co-Chair
Shiwei Lan, Co-Chair
Robert McCulloch

ARIZONA STATE UNIVERSITY

August 2023

ABSTRACT

Imitation Learning, also known as Learning from Demonstration (LfD), is a field of study dedicated to aiding an agent's learning process by providing access to expert demonstrations. Within LfD, Movement Primitives is a particular family of algorithms that have been widely studied and implemented in complex robot scenarios. Interaction Primitives, a subset of Movement Primitives, have demonstrated notable success in learning single interactions between humans and robots. However, literature addressing the extension of these algorithms to support multiple variations of an interaction is limited.

This thesis presents a physical human-robot interaction algorithm capable of predicting appropriate robot responses in complex interactions that involve a superposition of multiple interactions. The proposed algorithm, Blending Bayesian Interaction Primitives (B-BIP), achieves responsive motions in complex hugging scenarios and can reciprocate and adapt to the motion and timing of a hug. B-BIP generalizes prior work, where the original formulation reduces to the particular case of a single interaction. The performance of B-BIP is evaluated through an extensive user study and empirical experiments. The proposed algorithm yields significantly better quantitative prediction error and more favorable participant responses concerning accuracy, responsiveness, and timing compared to existing state-of-the-art methods.

ACKNOWLEDGMENTS

I would like to thank Dr. Heni Ben Amor for all of the support, guidance, and opportunities you have provided me with during my time at ASU. I am extremely fortunate to have been advised by a professor who is so compassionate, helpful, and intelligent. I would like to thank my committee members, Dr. Lan and Dr. McCulloch, and the professors in the School of Mathematical and Statistical Sciences for the invaluable learning experiences. I would like to express my gratitude to my family and friends for their unwavering support throughout my academic journey at ASU. I am immensely thankful to the Interactive Robotics Lab for the incredible support and opportunities provided to me. The lab has been an incubator for cherished friendships and invaluable research experiences that have exceeded all of my expectations. I have learned that research often involves building upon the work of those who came before us. I am particularly grateful to Joe Campbell for all of the knowledge, advice, and patience you've provided me with throughout my research journey. I am also fortunate to have collaborated with an outstanding team of researchers, including Simon, Geoff, Shubham, Xiao, Yifan, and Fabian (among many others), and I express my gratitude for their contributions to my growth. I would like to extend a special thank you to the Intrinsic team- Dr. Schaal, Ajinkya, Ravi, and Siva— for helping me with a project that has enriched my experiences in Imitation Learning. Finally, I wish to thank all the professors who have played a part in my academic and research development, including Dr. Artemiadis and professors from CIDSE (now SCAI). While there are undoubtedly others who have contributed to my journey, I acknowledge that without the collective support of everyone mentioned, my thesis would not be complete.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

Chapter 1

INTRODUCTION

Imitation learning (also known as LfD) is a field of study that involves training an agent to perform a certain task using expert demonstrations (Atkeson and Schaal, 1997; Pomerleau, 1988; Ross *et al.*, 2011; Ho and Ermon, 2016). The agent is typically a real-life robot or simulated entity with a set of actions. A closely related field, Reinforcement Learning (RL), also seeks to train an agent to learn how to perform tasks, but in RL, the agent learns by receiving feedback from the environment in the form of a reward signal (Kober *et al.*, 2013; Sutton and Barto, 2018; Bertsekas, 2019). Defining a reward signal can be difficult, especially in complex environments, which can lead to undesirable agent behaviors when not defined properly. RL algorithms can also demand significant computational efforts, especially in the case where the agent's action space is large and continuous.

This thesis presents an imitation learning algorithm inspired by the work done in Interaction Primitives (Amor *et al.*, 2014; Campbell and Ben Amor, 2017). The methodology is based on the work done by Drolet *et al.* (2023) and involves mathematical and statistical algorithms that form the foundation of the approach. These algorithms are applied to real-world human-robot interactions for evaluation, and the results can be extended to other interactions beyond human-robot hugging scenarios. As data-driven approaches become increasingly important in our world, algorithms that leverage prior expert demonstrations can be beneficial for learning an action policy that performs well in complex and noisy environments.

Chapter 2

PROBLEM FORMULATION

Learning from expert demonstrations is a powerful technique for training robots to perform complex tasks that would otherwise require significant engineering effort to manually design. In the context of a human-robot hugging scenario, learning from demonstration becomes even more important, as the nuances of human interactions and preferences can be difficult to capture through traditional engineering approaches. While one might consider using- for example- heuristics and simple feed-forward control to design a solution for hugging, such an approach would be highly specific to the particular scenario and would not generalize well to other interactions or scenarios. Furthermore, creating customized solutions for each interaction quickly becomes impractical due to the arbitrary number of observation features and degrees of freedom involved.

By contrast, learning the hugging interaction from expert demonstrations provides a clearly defined objective: to match the expert behavior as closely as possible. The expert trajectories contain nuanced and valuable information about the unique motions of agents, which may not be captured by traditional engineering approaches. Additionally, by allowing the expert to consist of demonstrations from multiple participants with varying heights, preferences, and proportions, the generated controls can adapt to a variety of situations naturally encoded by this distribution of expert trajectories. This learning-based approach provides a general, platform-agnostic solution that can be extended to new situations.

Implementing such behaviors in robots is an extremely challenging task and is often circumvented in experimental hugging robots by not reciprocating hugs (Yamazaki *et al.*, 2016; Sumioka *et al.*, 2013) or executing non-adaptive, pre-defined motions (Block and Kuchenbecker, 2019; Shiomi *et al.*, 2017; Hedayati *et al.*, 2019). Adaptive hugs require a robot to anticipate the type of hug performed, the current temporal progress, and the upcoming motion. Consequently, it has to generate accurate motor behavior to produce a synchronized motion with the human partner. One challenge in this regard is that hugs can be initiated anytime. Hence, the starting point is not predetermined nor easily identifiable. Another challenge is that hugs are typically fast movements with a duration of only a few seconds. Therefore, we need algorithms that allow robots to repeatedly (a) replan their motions in response to (b) the predicted behavior of the human partner. The proposed algorithm entailed in this thesis, B-BIP, aims to address these challenges by maintaining a list of possible interactions and fluidly transitioning between them in response to the interaction partner.

Traditional HRI methods classify the interaction first and then proceed with that classification throughout the interaction (Amor *et al.*, 2014; Ewerton *et al.*, 2015), or re-classify the interaction at discrete time intervals. This results in stilted HRI with limited capability to transition smoothly between actions or account for interactions that involve multiple actions. To overcome this, B-BIP allows for multiple interactions. At each time step, the likelihood of possible interactions is evaluated based on observations of the human partner. The models associated with each possible interaction are then updated based on the observations, with the magnitude of the update weighted by the likelihood. The intuition here is straightforward: if we are observing a seemingly unlikely interaction, then we do not want to update the model significantly because the observation is unlikely to have been generated from the model. It would only serve to distort it. Aside from being able to responsively transition from one interaction to another at any point in time, this approach has two subtle advantages: a) by updating all likely interactions at every time step, it avoids a sudden discontinuous transition between discrete interactions when switching occurs, and b) it is possible that interaction is a blend of multiple discrete interactions.

The main contributions of this thesis include a generalized form of Bayesian Interaction Primitives (Campbell and Ben Amor, 2017) that removes the restriction that a single interaction consists of only a single action, a probabilistic formulation where action transitions are detected to update a Monte-Carlo state approximation, and an empirical study demonstrating the successful detection and transitioning between three discrete sub-actions in a physical hugging scenario, compared to baseline methods. B-BIP's ability to adapt to a variety of interaction scenarios and seamlessly transition between different actions makes it a valuable tool in the design and implementation of robot behaviors for real-world interactions.

Chapter 3

PRELIMINARIES

The following subsections cover the preliminary methods necessary for the implementation and understanding of the main work in this thesis.

3.1 Kalman Filtering

The Kalman filter (Kalman, 1960; Thrun *et al.*, 2005) is a powerful tool for estimating the state of a system, especially when dealing with noisy measurements. In the field of robotics, it is commonly used for localization, which is the process of determining the position and orientation of a robot in an environment. For this work, the Kalman filter is used to estimate a latent state which, when projected back to the observation space, reconstructs the full trajectories of the human and robot. To motivate the use the Kalman filter for state estimation, consider a simple 2D constant velocity model. The state representation $\boldsymbol{x}_t$ consists of the position (x_t, y_t) and the velocities (v_{x_t}, v_{y_t}) of the robot at time t. The state transition function is given by $\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1}$, and the matrices involved can be defined as:

- State transition matrix, $\boldsymbol{F}_t$:

$$\boldsymbol{F}_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

- State vector, $\boldsymbol{x}_t$:

$$\boldsymbol{x}_t = \begin{bmatrix} x_t \\ y_t \\ v_{x_t} \\ v_{y_t} \end{bmatrix} \tag{3.2}$$

where Δt is the time step. Along with the state transition update, the covariance associated with the state vector is updated with an additive process noise.

- Process noise covariance matrix, $\boldsymbol{Q}_t$:

$$\boldsymbol{Q}_t = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \tag{3.3}$$

- State covariance matrix, $\boldsymbol{\Sigma}_t$:

$$\boldsymbol{\Sigma}_t = \boldsymbol{F}_t \boldsymbol{\Sigma}_{t-1} \boldsymbol{F}_t^T + \boldsymbol{Q}_t \tag{3.4}$$

Here, q_1, q_2, q_3, and q_4 are the variances for the process noise in the respective state variables. The equations above define a linear dynamical model for updating the state estimate recursively. However, this is a simple (first order) approximation of the system dynamics which does not account for external perturbations; as such, this leads to compounding errors in a non-idealized setting. To correct the transition update, we can take advantage of sensor measurements. Often sensors come with a reference sheet that specifies the precision/uncertatinty associated with the sensor, and these values can be used to instantiate a measurement noise matrix that is representitive of the true sensor uncertainty. The following sub-section will introduce the machinery for how this update step is performed. The subscripts for the state variables are replaced with a more descriptive notation used to indicate the time step with which the variable is conditioned on.

3.1.1 Linear Kalman Filter

The Kalman filter algorithm consists of two main steps: the prediction step and the update step. The equations that govern these steps are presented below.

Prediction step:

$$\boldsymbol{x}_{t|t-1} = \boldsymbol{F}_t \boldsymbol{x}_{t-1|t-1} + \boldsymbol{B}_t \boldsymbol{u}_t \tag{3.5}$$

$$\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{F}_t \boldsymbol{\Sigma}_{t-1|t-1} \boldsymbol{F}_t^\top + \boldsymbol{Q}_t \tag{3.6}$$

Update step:

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{H}_t^\top (\boldsymbol{H}_t \boldsymbol{\Sigma}_{t|t-1} \boldsymbol{H}_t^\top + \boldsymbol{R}_t)^{-1} \tag{3.7}$$

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_{t|t-1} + \boldsymbol{K}_t (\boldsymbol{y}_t - \boldsymbol{H}_t \boldsymbol{x}_{t|t-1}) \tag{3.8}$$

$$\boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t) \boldsymbol{\Sigma}_{t|t-1} \tag{3.9}$$

Here, $\boldsymbol{B}$ is the control input matrix which, depending on the context, need not always be implemented (as in the motivating example); $\boldsymbol{K}$ is the Kalman gain matrix; $\boldsymbol{H}$ is the measurement matrix; $\boldsymbol{R}$ is the measurement noise covariance matrix; and $\boldsymbol{y}$ is the measurement vector. The process noise covariance matrix and measurement noise covariance matrix are defined by the user in the case of B-BIP.

3.1.2 Optimal Gain

The motivation for using the Kalman filter comes from the idea that it is optimal from a least-squares perspective (Gelb *et al.*, 1974). As we will see, the Kalman Gain minimizes the square of the updated state covariance at time t. Consequently, the specific choice of the Kalman gain in the state update equation above can be attributed to minimizing the uncertainty of the state estimate. To show why this holds, we first start with the Joseph form of the covariance update equation, which can be derived from the posterior estimate of the covariance matrix as done in Benhamou (2018):

$$\boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}_t)\boldsymbol{\Sigma}_{t|t-1}(\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}_t)^\top + \boldsymbol{K}_t\boldsymbol{R}_t\boldsymbol{K}_t^\top \tag{3.10}$$

This is true for any value of $\boldsymbol{K}_t$, so this equation can be used to solve for the optimal Kalman gain. Expanding the terms, we get:

$$\boldsymbol{\Sigma}_{t|t} = \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{K}_t\boldsymbol{H}_t\boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^\top\boldsymbol{K}_t^\top + \boldsymbol{K}_t\boldsymbol{S}_t\boldsymbol{K}_t^\top \tag{3.11}$$

where $\boldsymbol{S}_t = \boldsymbol{H}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^\top + \boldsymbol{R}_t$. Next, we take the derivative of the trace of the state covariance matrix with respect to the Kalman gain. Setting this expression equal to zero allows us to solve for the value of $\boldsymbol{K}_t$ that minimizes the state uncertainty (i.e, the sum of the variances).

$$\frac{\delta \mathrm{tr}(\boldsymbol{\Sigma}_{t|t})}{\delta \boldsymbol{K}_t} = -2(\boldsymbol{H}_t\boldsymbol{\Sigma}_{t|t-1}^\top)^\top + 2\boldsymbol{K}_t\boldsymbol{S}_t = 0 \tag{3.12}$$

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^\top\boldsymbol{S}_t^{-1} \tag{3.13}$$

$$= \frac{\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^\top}{\boldsymbol{H}\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^\top + \boldsymbol{R}_t} \tag{3.14}$$

We can view the Kalman gain as a ratio (or compromise) of the *state uncertainty* to the *state and measurement uncertainty*. When $\boldsymbol{\Sigma}_{t|t-1}$ is fixed and $\boldsymbol{R}_t$ takes on higher values (e.g, when the sensors are noisier), then the Kalman gain becomes smaller.

3.1.3 Ensemble Kalman Filter

The ensemble Kalman filter is a Monte-Carlo approximation of the Kalman filter, originally introduced by Evensen (2003). It's important to note that in ensemble Kalman filter, the prediction step and update step are similar to those of the standard Kalman filter. The main difference is that in the ensemble Kalman filter, instead of working with a single estimate of the state and its covariance, we work with a set of E samples (or ensemble members) that approximate the state distribution.

The ensemble Kalman filter can also improve computational efficiency as it does not require maintaining a covariance matrix at every iteration, which can be computationally expensive for high-dimensional systems. This makes the ensemble Kalman filter suitable method for high-dimensional systems where the traditional Kalman filter may struggle with computational complexity. Below, the transition model and observation model, denoted by $f(.)$ and $h(.)$ respectively, are used instead of the matrix notation above. The superscript j is used to denote the j'th ensemble member (a state vector) such that when j is omitted, then the whole ensemble (a matrix) is referred to.

Prediction Step:

$$\boldsymbol{X}^{j}_{t|t-1} = f(\boldsymbol{X}^{j}_{t-1|t-1}) + \boldsymbol{\epsilon}^{j}_{\boldsymbol{Q}_t} \quad \text{s.t.} \quad \boldsymbol{\epsilon}^{1}_{\boldsymbol{Q}_t}, \cdots, \boldsymbol{\epsilon}^{E}_{\boldsymbol{Q}_t} \overset{\text{i.i.d}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_t) \tag{3.15}$$

Update Step:

$$[\boldsymbol{H}_t\boldsymbol{A}_t]^j = h(\boldsymbol{X}^j_{t|t-1}) - \frac{1}{E}\sum_{j=1}^{E} h(\boldsymbol{X}^j_{t|t-1}) \tag{3.16}$$

$$\boldsymbol{A}^j_t = \boldsymbol{X}^j_{t|t-1} - \frac{1}{E}\sum_{j=1}^{E} \boldsymbol{X}^j_{t|t-1} \tag{3.17}$$

$$\boldsymbol{S}_t = \frac{1}{E-1}(\boldsymbol{H}_t\boldsymbol{A}_t)(\boldsymbol{H}_t\boldsymbol{A}_t)^\top + \boldsymbol{R}_t \tag{3.18}$$

$$\boldsymbol{K}_t = \frac{1}{E-1}\boldsymbol{A}_t(\boldsymbol{H}_t\boldsymbol{A}_t)^\top \boldsymbol{S}_t^{-1} \tag{3.19}$$

$$\tilde{\boldsymbol{Y}}^j_t = \boldsymbol{y}_t + \boldsymbol{\epsilon}^j_{\boldsymbol{R}_t} \quad \text{s.t.} \quad \boldsymbol{\epsilon}^1_{\boldsymbol{R}_t}, \cdots, \boldsymbol{\epsilon}^E_{\boldsymbol{R}_t} \overset{\text{i.i.d}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_t) \tag{3.20}$$

$$\boldsymbol{H}_t\boldsymbol{X}_{t|t-1} = \left[h(\boldsymbol{X}^1_{t|t-1}), \cdots, h(\boldsymbol{X}^E_{t|t-1})\right] \tag{3.21}$$

$$\boldsymbol{X}_{t|t} = \boldsymbol{X}_{t|t-1} + \boldsymbol{K}_t(\tilde{\boldsymbol{Y}}_t - \boldsymbol{H}_t\boldsymbol{X}_{t|t-1}) \tag{3.22}$$

In the equations above, the process noise covariance $\boldsymbol{Q}_t$ and the measurement noise covariance $\boldsymbol{R}_t$ are defined by the user. We can see how the sample covariance is used in the Kalman gain update, i.e, $\frac{\boldsymbol{A}\boldsymbol{A}^\top}{E-1}$, as opposed to maintaining a covariance as in the linear Kalman filter. Another clear difference from the standard Kalman filter is that the observation function $h(.)$ and transition function $f(.)$ are applied to each ensemble member iteratively. These functions may be nonlinear, as the update is no longer confined to matrix operations where linearization is applied. Additionally, noise is applied to the measurement $\boldsymbol{y}_t$ to treat the measurement as random and give the ensemble a proper variance (Burgers *et al.*, 1998).

3.2 BIP

3.2.1 Basis Function Decomposition

First, a primitive is trained using a set of demonstrations of the desired interaction. Each demonstration consists of *observed* and *controlled* degrees of freedom, which are modeled as the matrix $\boldsymbol{Y} \in \mathbb{R}^{T \times D}$ where $D = |D_c| + |D_o|$ denotes the total number of degrees of freedom (DoFs) in the interaction (having the sets of DoF indices D_c from the controlled agent and D_o DoFs from the observed agent) and T samples.

These demonstrations are then transformed to a time-invariant latent space $\boldsymbol{w}$ such that

$$\boldsymbol{y}_t^d = h^d(\phi(t), \boldsymbol{w}^d) = \boldsymbol{\Phi}_{\phi(t)}^d \boldsymbol{w}^d + \epsilon_y, \tag{3.23}$$

where $\boldsymbol{\Phi}_{\phi(t)}^d \in \mathbb{R}^{1 \times B^d}$ is a row vector of B^d basis functions, $\boldsymbol{w}^d \in \mathbb{R}^{B^d \times 1}$, and ϵ_y is i.i.d. Gaussian noise. As is standard (Campbell and Ben Amor, 2017), the basis functions are dependent on a relative time measure referred to as phase $\phi(t)$. The Gaussian basis function is used in this work, although others may be selected if appropriate for the task domain. The basis functions for a given DoF d are evaluated on a discrete, linearly-spaced interval of length T from $\phi(1) = 0$ to $\phi(T) = 1$. For a given phase ϕ, the Gaussian basis function evaluations produce a row vector:

$$\boldsymbol{\Phi}_\phi^d = \begin{bmatrix} \eta \exp\left(-h(\phi - c_1)^2\right), & \eta \exp\left(-h(\phi - c_2)^2\right), & \cdots, & \eta \exp\left(-h(\phi - c_{B^d})^2\right) \end{bmatrix} \tag{3.24}$$

where h represents a shared scale (for simplicity), c_i is the center of the basis function, and η is a normalizing constant. These row vectors are stacked to form the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{T \times B^d}$ used to reconstruct the full trajectory $\boldsymbol{y}^d \in \mathbb{R}^{T \times 1}$. The normal equation for reconstructing the full trajectory is given by:

$$\boldsymbol{w}^d = \left(\boldsymbol{\Phi}^{d\top}\boldsymbol{\Phi}^d\right)^{-1}\boldsymbol{\Phi}^{d\top}\boldsymbol{y}^d \tag{3.25}$$

A regularization term can also be added when solving for the weights. Paraschos *et al.* (2018) show that this term can also be modified to minimize jerk. Below is the addition of a ridge regression penalty term:

$$\boldsymbol{w}^d = \left(\boldsymbol{\Phi}^{d\top}\boldsymbol{\Phi}^d + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^{d\top}\boldsymbol{y}^d \tag{3.26}$$

The ridge regression estimate can be obtained as the MAP estimate of a Bayesian linear regression model with a zero-mean Gaussian prior on the weights (O'Hagan and Forster, 2004). The ridge regression estimate corresponds to the mean of the posterior distribution, while the L2 penalty controls the variance of the posterior distribution. In this sense, ridge regression can be seen as a Bayesian approach to regularization (Toussaint and Goerick, 2010). We are influencing the weights to be smaller through the zero-mean prior term; importantly, shrinking these weights can result in desirable properties such as tightening the range of the high-dimensional state space. Alternatively, using a lasso penalty (L1 regularization) for model selection could help prune any redundant basis functions if they are over-specified.

During inference, we wish to estimate the latent state $\boldsymbol{w}$ from which the inferred controlled DoFs can be retrieved, however, it is also necessary to localize both the phase, ϕ, and phase velocity, $\dot{\phi}$, in order to accurately perform inference on demonstrations done at different speeds. Therefore, the state representation is augmented to be

$$\boldsymbol{s}_t = [\boldsymbol{\phi}_t, \dot{\boldsymbol{\phi}}_t, \boldsymbol{w}_t]. \tag{3.27}$$

This state representation was introduced in the original BIP work, which makes an important contribution to the setup of the system by estimating the phase of the interaction jointly. This work shows that the temporal estimation error induces correlated spatial errors, which can be analogous to Simultaneous Localization and Mapping (SLAM).

3.2.2 State Space Filtering

Using Bayes rule, the posterior state estimate can be obtained by using the previous state estimate and the likelihood of the measurement. Given a sequence of measurements, $\boldsymbol{Y}_{1:t}$, of all modalities, this gives us the following probabilistic formulation:

$$p(\boldsymbol{s}_t|\boldsymbol{Y}_{1:t}, \boldsymbol{s}_0) \propto p(\boldsymbol{y}_t|\boldsymbol{s}_t)p(\boldsymbol{s}_t|\boldsymbol{Y}_{1:t-1}, \boldsymbol{s}_0). \tag{3.28}$$

As in Campbell *et al.* (2019b), this posterior density is approximated using a Monte Carlo method in which an ensemble of samples are used to predict the next measurement,

$$\boldsymbol{X}^j_{t|t-1} = f(\boldsymbol{X}^j_{t-1|t-1}) + \boldsymbol{\epsilon}^j_{\boldsymbol{Q}_t} \quad \text{s.t.} \quad \boldsymbol{\epsilon}^1_{\boldsymbol{Q}_t}, \cdots, \boldsymbol{\epsilon}^E_{\boldsymbol{Q}_t} \overset{\text{i.i.d}}{\sim} \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{Q}_t\right) \tag{3.29}$$

The predicted ensemble is updated based on the error obtained from the actual measurement using a gain coefficient $\boldsymbol{K}_t$, as shown in the EnKF overview section.

$$\boldsymbol{X}_{t|t} = \boldsymbol{X}_{t|t-1} + \boldsymbol{K}_t(\tilde{\boldsymbol{Y}}_t - \boldsymbol{H}_t\boldsymbol{X}_{t|t-1}) \tag{3.30}$$

The set of training demonstrations that we start with are used to directly initialize the ensemble members, such that E demonstrations yields E ensemble members.

This procedure follows the ensemble Kalman filter algorithm detailed in Section 3. While the Kalman filter is commonly used for modeling a non-latent state space, such as positions and velocities, BIP distinguishes itself by adopting a different approach. Instead, BIP's state space consists of basis function weights that form the latent state.

The measurement function, $h(\boldsymbol{x}^j_{t|t-1})$ transforms this latent state into the measurement space by applying the basis function matrix (at the given phase) to the weights, such that the residual from the observation can then be computed. B-BIP builds off of this work by dividing the state space into discrete sub-ensembles, where we are then able to update each filter according to the likelihood of the interaction class.

3.3 Discriminant Analysis

Quadratic Discriminant Analysis (QDA) and Linear Discriminant Analysis (LDA) are commonly employed techniques in classification (Hastie *et al.*, 2009). Both methods seek to identify decision boundaries that optimize the separation between classes. QDA is an extension of LDA, permitting the covariance of each class to differ, whereas LDA presumes that the covariance of every class is identical. The QDA model assumes that the data for each class follows a normal distribution with distinct covariance matrices, denoted as $\boldsymbol{\Sigma}_i$. Given a new observation $\boldsymbol{x}$, the class membership probability for class k is expressed by:

$$\hat{y} = \arg\max_k \left\{ -\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \ \boldsymbol{\Sigma}_k^{-1} \ (\boldsymbol{x} - \boldsymbol{\mu}_k) + \log\pi_k \right\} \tag{3.31}$$

On the other hand, LDA assumes that the data for each class follows a normal distribution with a consistent covariance matrix, denoted as $\boldsymbol{\Sigma}$. Under LDA, the class membership probability for class k is given by:

$$\hat{y} = \arg\max_k \left\{ \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \log\pi_k \right\} \tag{3.32}$$

A primary advantage of LDA is its capability to compute a shared subspace for all classes, facilitating dimensionality reduction. This is not the case with QDA, as it does not make this assumption and thus does not calculate a common subspace for all classes. Section 5.2 provides a demonstration of how LDA is employed within Blending Bayesian Interaction Primitives to effectively reduce the dimensionality of the demonstration data and maintain the discriminative information essential for blending human-robot interactions.

Chapter 4

RELATED WORK

This section will give an overview of existing methods in imitation learning that are most similar to this work. In particular, Movement Primitives in general have a well-established history, starting with Dynamic Movement Primitives (DMPs). The LSTM (Hochreiter and Schmidhuber, 1997) is a general method used for predicting time series data and it is adopted as a baseline for comparison; as such, it is not restricted to imitation learning.

4.1 DMPs

Dynamic Movement Primitives are a popular framework for modeling and generating movement trajectories (Ijspeert *et al.*, 2013; Schaal, 2006). DMPs were inspired by the concept of motor primitives, which bear biological connections to humans and animals. DMPs provide a principled, control-theoretic approach to encode essential features of a movement, allowing them to accurately reproduce the unique action trajectories exhibited by robots, humans, or other agents. A DMP consists of a canonical system that acts as a clock for progressing the phase of the movement trajectory and a forcing term is responsible for shaping the trajectory toward a desired goal. A dynamical system is formulated as a damped spring model of the form:

$$\tau\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f, \tag{4.1}$$

where y is the state of the system, τ is the time constant that determines the speed of the movement, g is the goal position, α_y and β_y are gain terms, and f is the forcing function that can be constructed using an expert trajectory. The dynamics of the canonical system are given by:

$$\dot{x} = -\alpha_x x. \tag{4.2}$$

The forcing term, a function of the canonical system, is formulated as a weighted sum of basis functions of the form:

$$f(x) = \frac{\sum_{i=1}^{N} \Phi_i(x) w_i}{\sum_{i=1}^{N} \Phi_i(x)} x(g - y_0), \tag{4.3}$$

where N is the number of basis functions and w_i is the weight of the i-th basis function.

The DMP can be executed in real-time by using the canonical system to generate the basis signal and the forcing term to shape the trajectory toward the goal. DMPs have been successfully applied to a wide range of robotic and human movement tasks, including reaching, grasping, and walking. One of the main advantages of DMPs is their ability to generate smooth and natural-looking movements that can easily be adapted to different goal locations.

4.2 ProMPs

The Probabilistic Movement Primitives (ProMPs) alorithm introduced by Paraschos *et al.* (2013) is a probabilistic extension of the Dynamic Movement Primitives (DMPs) framework. Like DMPs, ProMPs provide a way to encode motion trajectories for robots in a low-dimensional and task-specific way. However, ProMPs go beyond the deterministic framework of DMPs by using a Gaussian distribution over the trajectory weights, allowing for recursive probabilistic inference and uncertainty quantification. While BIP differs from DMPs and ProMPs in some key aspects such as in phase localization and filtering (via the extended and ensemble Kalman filter), it utilizes the same concepts related to basis function decomposition and weight space representation presented in this prior work.

4.3 LSTM Networks

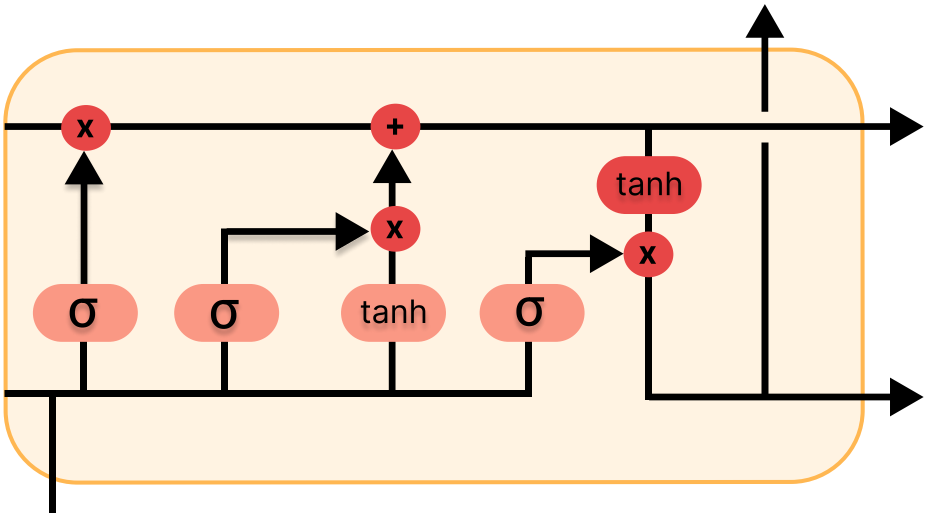

Figure 1. Architecture of LSTM Cell

The Long-Short Term Memory Network (LSTM) is popular for its ability to model and predict sequential data, particularly in the realm of time series analysis. This can be attributed to the LSTM's unique architecture, which facilitates the "forgetting" and "remembering" of past information to enhance the prediction of subsequent states. The LSTM's recurrent architecture is characterized by a series of interconnected memory cells, wherein each cell's estimate of the previous state is supplied as input to the subsequent cell. This iterative process allows the network to learn and maintain information over extended sequences, effectively overcoming the limitations of traditional feedforward networks in modeling long-range dependencies.

In spite of the complexity and sophistication of the LSTM's underlying mechanisms, its implementation in real-world applications has been made considerably more accessible through the availability of user-friendly software libraries. These libraries provide a wealth of tools and resources that allow researchers and practitioners to develop and deploy LSTM-based models with relative ease, streamlining the process of adapting the network architecture to the specific requirements of the task at hand.

However, it is important to recognize that the performance of an LSTM model is heavily contingent upon the configuration of its network architecture. The selection of appropriate hyperparameters, such as the number of layers, hidden units, and activation functions, plays a crucial role in determining the model's capacity to effectively learn and generalize from the given data. Consequently, it is often necessary to engage in a process of hyperparameter tuning and optimization to identify the optimal network settings for a particular problem.

Chapter 5

METHODOLOGY

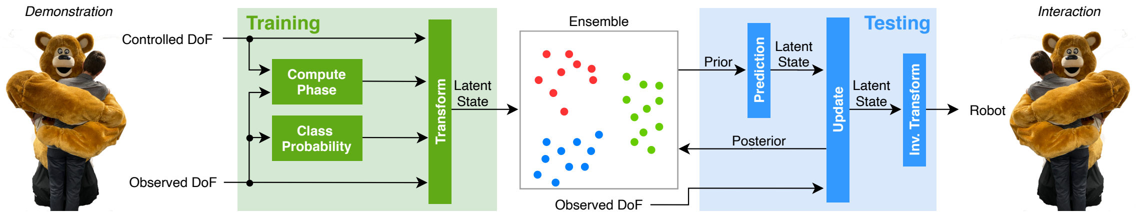

Figure 2. An overview of B-BIP. Top: training demonstrations (left) are decomposed into a latent space (middle) and transformed into an ensemble of samples (right). Bottom: observations are collected during a live interaction (left) which is used to perform filtering with the learned ensemble (middle) and produce a response trajectory (right).

5.1 Ensemble Partitioning

The BIP framework as described above only supports modeling a single interaction at a time. In order to extend this model to a set of interactions, we first present a probabilistic formulation for what this entails. Suppose we have an ensemble $\boldsymbol{X}$ in which each of the E ensemble members belongs to a class $c \in \mathcal{C}$, in which the set of classes $\mathcal{C}$ represents different discrete interactions. This allows us to partition $\boldsymbol{X}$ into $|\mathcal{C}|$ sub-ensembles, such that $\boldsymbol{X} = \boldsymbol{X}^{(1)} \cup \boldsymbol{X}^{(2)} \cup \cdots \cup \boldsymbol{X}^{(|\mathcal{C}|)}$ where a sub-ensemble $\boldsymbol{X}^{(c)}$ contains $E^{(c)}$ members such that $E = \sum_{c \in \mathcal{C}} E^{(c)}$. We define C as a random variable over the set $\mathcal{C}$ which indicates the class of an interaction. Each sub-ensemble $\boldsymbol{X}^{(c)}$ is a Monte Carlo approximation of the probability distribution for the c-th interaction, $p(\boldsymbol{s}_t | \boldsymbol{Y}_{1:t}, \boldsymbol{s}_0, C = c)$, for which we can marginalize out C to re-obtain the full posterior distribution:

$$p(\boldsymbol{s}_t|\boldsymbol{Y}_{1:t}, \boldsymbol{s}_0) = \sum_{c \in \mathcal{C}} p(\boldsymbol{s}_t|\boldsymbol{Y}_{1:t}, \boldsymbol{s}_0, C = c) p(C = c|\boldsymbol{Y}_{1:t}, \boldsymbol{s}_0). \tag{5.1}$$

The association of each ensemble member to a class c is static and defined in the prior distribution $\boldsymbol{s}_0$, as demonstrations must be initially collected for each individual interaction and hence we have a mapping from demonstrations to classes. This allows us to calculate the posterior for a specific class, $p(\boldsymbol{s}_t|\boldsymbol{Y}_{1:t}, \boldsymbol{s}_0, C = c)$, in a similar manner as Eqs. 3.29-3.30 but limited to the ensemble members $\boldsymbol{x} \in \boldsymbol{X}^{(c)}$; this is covered in Sec. 5.3. We do not restrict ourselves to the case that C is fixed to a single value c; an interaction may transition between multiple classes over time. Therefore, the interaction scenarios examined in previous works are special cases of this formulation and only take on one class value.

5.2 Interaction Classification

In this section, a Reduced-Rank Linear Discriminant Analysis (LDA) method is proposed for computing the probability of an interaction class, given the current observations of a human. The method is based on the concept of within-class and between-class scatter matrices, which are used to perform dimensionality reduction on the demonstration data.

Given a set of training demonstrations, $\boldsymbol{Y}_i^{(c)} \in \mathbb{R}^{D \times T}$, for each interaction class $c \in \mathcal{C}$, where $1 \leq i \leq N^{(c)}$, and $N^{(c)}$ denotes the total number of demonstrations for class c. The within-class demonstration matrix is defined as

$$\boldsymbol{M}^{(c)} = [(\boldsymbol{Y}_1^{(c)})_{D_o,:} \ , \ \dots \ , \ (\boldsymbol{Y}_{N^{(c)}}^{(c)})_{D_o,:}] \tag{5.2}$$

and the between-class demonstration matrix is defined as $\boldsymbol{M} = \left[\boldsymbol{M}^{(1)}, \dots, \boldsymbol{M}^{(|\mathcal{C}|)}\right]^\top$.

Multiple Discriminant Analysis is used to find a low-rank representation of $\boldsymbol{M}$ and perform dimensionality reduction by computing the solution of the Rayleigh coefficient, which is the ratio of the between-class scatter to within-class scatter. The within-class scatter matrix is defined as the sum of prior-weighted class covariance matrices:

$$\boldsymbol{S}_W = \sum_{c \in \mathcal{C}} \left[\sum_{i=1}^{sN^{(c)}} \frac{(\boldsymbol{M}_{:,i}^{(c)} - \boldsymbol{\mu}^{(c)})(\boldsymbol{M}_{:,i}^{(c)} - \boldsymbol{\mu}^{(c)})^\top}{N^{(c)} - 1} \right] \pi^{(c)}. \tag{5.3}$$

The total scatter matrix is defined as:

$$\boldsymbol{S}_T = \sum_{\boldsymbol{m} \in \text{rows}(\boldsymbol{M})} (\boldsymbol{m} - \boldsymbol{\mu})(\boldsymbol{m} - \boldsymbol{\mu})^\top, \tag{5.4}$$

where $\boldsymbol{\mu}$ is the total mean vector (defined as the mean over all measurements in the space) and $\boldsymbol{m}$ is a measurement from the set of all measurements in the space (i.e, for every row in the $\boldsymbol{M}$ matrix). The between-class scatter matrix is defined as $\boldsymbol{S}_B = \boldsymbol{S}_T - \boldsymbol{S}_W$. The s in the summation for $\boldsymbol{S}_W$ represents a (fixed) randomly selected number of samples taken from a training demonstration. By randomly sampling s points from each demonstration instead of using the full number of samples for each demonstration, the computational complexity for the scatter matrix is greatly reduced. Considering the measurements are collected at 120 Hz, sub-sampling the trajectories does not provide any meaningful loss in information. The eigenvectors, $\boldsymbol{w}_i$, of the transformation matrix, $\boldsymbol{W}$, that maximize the ratio of between-class scatter to within-class scatter are obtained by solving $(\boldsymbol{S}_B - \lambda_i \boldsymbol{S}_W)\boldsymbol{w}_i = 0$.

The eigenvectors corresponding to the $k = |\mathcal{C}| - 1$ largest eigenvalues are selected to construct the reduced-rank representation of the transformation matrix $\boldsymbol{W}_k$. The distribution of samples are subsequently projected onto the k-dimensional subspace spanned by $\boldsymbol{W}_k$. For notational simplicity, $\boldsymbol{Z} = \boldsymbol{W}_k^\top \boldsymbol{M}$ and $\boldsymbol{z}_t = \boldsymbol{W}_k^\top \boldsymbol{y}_t^{D_o}$ are defined.

Finally, the posterior density is computed, taking into account the previous observations and the class prior probability. The resulting form is:

$$\begin{aligned}\log p(C = c \mid \boldsymbol{Y}_{1:t}, \boldsymbol{s}_0) &= \log p(\boldsymbol{Y}_{1:t}, \boldsymbol{s}_0 \mid C = c) + \log p(C = c) + \eta \\ &= -\frac{1}{2}(\boldsymbol{z}_t - \boldsymbol{\mu}_{\boldsymbol{Z}^{(c)}})^\top \Sigma_{\boldsymbol{Z}}^{-1}(\boldsymbol{z}_t - \boldsymbol{\mu}_{\boldsymbol{Z}^{(c)}}) + \log p(C = c) + \eta. \quad (5.5)\end{aligned}$$

After dropping the quadratic term $\boldsymbol{z}_t^\top \Sigma_{\boldsymbol{Z}}^{-1} \boldsymbol{z}_t$ from 5.5, which is independent of c, we get the resulting form

$$\log p(C = c \mid \boldsymbol{Y}_{1:t}, \boldsymbol{s}_0) = \boldsymbol{\beta}_c^\top \boldsymbol{z}_t + \beta_{c0}. \quad (5.6)$$

Here, $\boldsymbol{\beta}_c = \Sigma_{\boldsymbol{Z}}^{-1} \boldsymbol{\mu}_{\boldsymbol{Z}^{(c)}}$ and $\beta_{c0} = -\frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{Z}^{(c)}}^\top \Sigma_{\boldsymbol{Z}}^{-1} \boldsymbol{\mu}_{\boldsymbol{Z}^{(c)}} + \log \pi^{(c)}$. The posterior can now be computed by applying the softmax function,

$$p(C = c \mid \boldsymbol{Y}_{1:t}, \boldsymbol{s}_0) = \frac{e^{\boldsymbol{\beta}_c^\top \boldsymbol{z}_t + \beta_{c0}}}{\sum_{j \in C} e^{\boldsymbol{\beta}_j^\top \boldsymbol{z}_t + \beta_{j0}}}. \quad (5.7)$$

5.3 Gain Weigthing

When calculating Eq. 5.1 we must be careful to weight the magnitude of the ensemble update with the class probability. The intuitive reason is that the standard ensemble member update of Eq. 3.30 assumes that the observation was generated from a distribution approximated by that ensemble. However, if there is only a small probability that the observation was generated from a particular sub-ensemble and we apply a full-magnitude update then we potentially skew the ensemble members with an out-of-distribution measurement. Thus, unlike Eq. 3.30, we now weight the gain coefficient with Eq. 5.7:

$$\boldsymbol{x}_{t|t}^j = \boldsymbol{x}_{t|t-1}^j + p(C = c \mid \boldsymbol{Y}_{1:t}, \boldsymbol{s}_0)\boldsymbol{K}(\tilde{\boldsymbol{y}}_t - h(\boldsymbol{x}_{t|t-1}^j)) \quad (5.8)$$

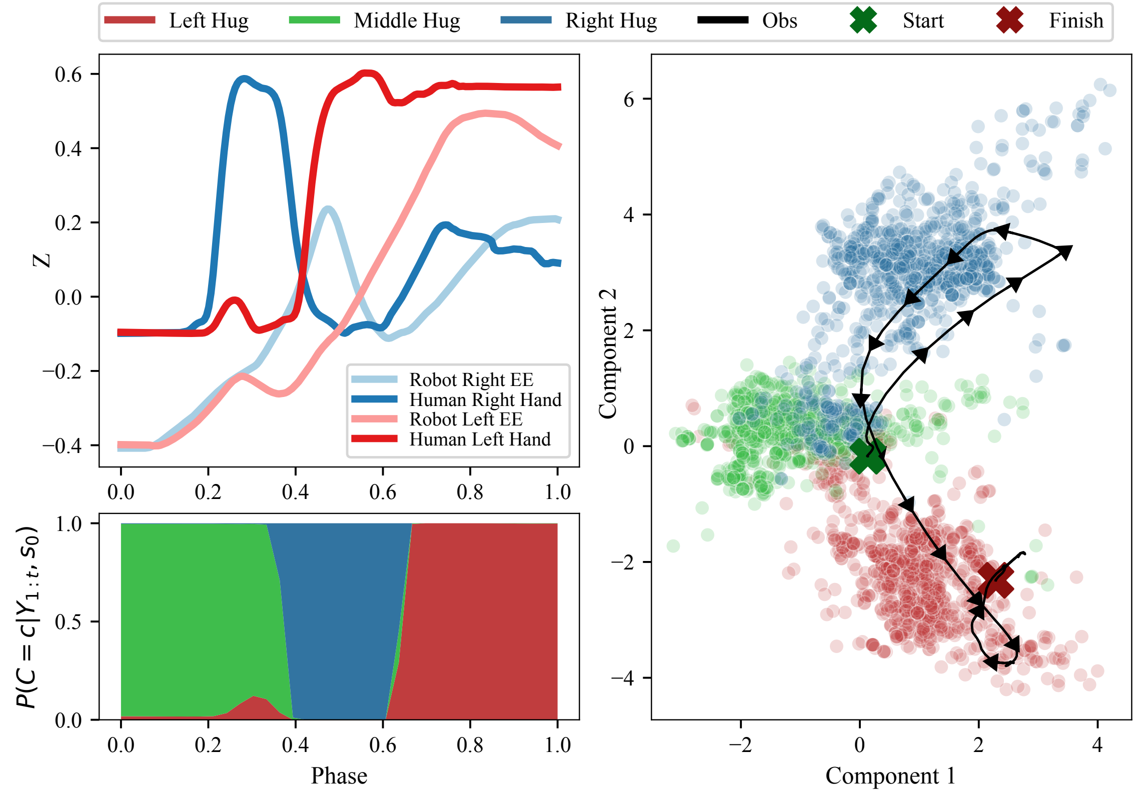

Figure 3. Top left: The observed z positions of the participant's hands and the robot's end effectors during a right-high to left-high interaction. Bottom left: The corresponding interaction class weights for each hug type, with vertical area equal to the class probability. Right: The trajectory of the observed DoF (black arrows) projected to the reduced-rank LDA space, overlaid on the distributions (circles) for each hug type.

Chapter 6

EXPERIMENTAL DESIGN

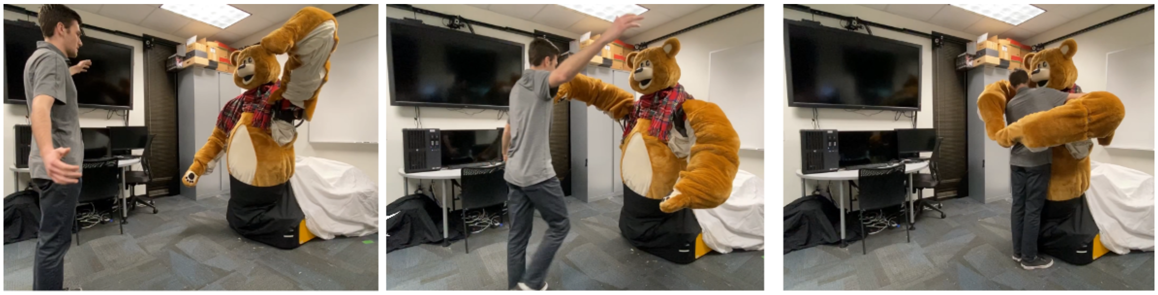

Figure 4. An example of a *left-high* to *right-high* interaction. Left: The participant starts with a left-high interaction. Middle: When switching to the right-high hug, the robot responds accordingly. Right: The participant hugs the robot.

In order to empirically evaluate B-BIP, an IRB-approved participant study is conducted which consists of a hugging scenario between a robot and a human.

6.1 Training Data Collection

A motion capture system is utilized to collect data at 120 Hz from participants wearing a hat and a wristband on each hand, providing a total of three observed modalities. Although the data contains the 3D positions and orientations for each modality, the orientation information is not used, as the orientation of the hands and head is not of interest during the hug interaction. A total of 15 demonstrations of each interaction type (left-high, middle, and right-high) are collected from 15 different participants, amounting to 225 demonstration hugs per interaction class and 675 in total. The term left-high is used to indicate an interaction where the left hand of the robot and the human are raised, such that the human's left hand approaches the robot over its right shoulder; the same symmetry holds for the right-high interaction, and middle is used to denote a hug where the robot hugs with arms at the same height (see Fig. 4). A response elicitation technique (Campbell *et al.*, 2019a) is applied, in which the robot actuates according to an open-loop control policy, and the human responds accordingly. Outlier training demonstrations with any DoF value outside of four standard deviations from the distribution of demonstrations at the same point in time are removed from the dataset. Subsequently, an 80-20 percent train-test split is employed to train and validate the methods for mean-squared error (MSE) comparison. This results in 439 training demonstrations and 110 validation demonstrations.

6.2 Prediction Methods

The goal of the study is to compare 1) B-BIP, 2) BIP, 3) Probabilistic Movement Primitives (ProMP) (Ewerton *et al.*, 2015; Maeda *et al.*, 2017; Paraschos *et al.*, 2013), and 4) a LSTM network, all of which are evaluated on *non-switching* interactions (left-high, middle, and right-high hugs) as well as on *switching* interactions (transitions from either left-high to right-high or right-high to left-high hugs). The LSTM architecture contains: 28 hidden units (twice the number of DOFs from the robot), a dropout layer with rate of 0.2, a batch normalization layer, a fully-connected layer with 28 units, and a fully-connected layer with 14 units at the output. The LSTM was trained using a sliding window with a queue data structure, wherein the last n (window size) timesteps are used to predict the next action. This setup allows for a computationally efficient way of predicting actions during inference, compared to using the full history at every timestep which becomes slower as time progresses. In the case of ProMP, phase estimation is performed via Dynamic Time Warping (DTW) as described in (Maeda *et al.*, 2017). All methods are trained on the 439 demonstrations previously described; the alternative methods treat each interaction as one class while Blending takes into account the interaction class labels.

6.3 Hypotheses

Three critical factors are identified for evaluation when comparing the B-BIP to alternative methods. After every hug, the following questions are posed:

1. On a scale of 1 to 5, how good was the timing of the robot during the interaction?
2. On a scale of 1 to 5, how well did the robot match your type of hug during the interaction?
3. On a scale of 1 to 5, how responsive was the robot to your motion?

The main hypotheses are stated below, applying to both non-switching and switching interactions:

- $\boldsymbol{H_1}$: Proposed method better matches the hug type than baseline methods.
- $\boldsymbol{H_2}$: Proposed method has better timing than baseline methods.
- $\boldsymbol{H_3}$: Proposed method elicits more responsive behavior than baseline methods.

6.4 Participant Study

The participant study was conducted on 22 new participants whose data was not used to train the methods, including 4 female and 18 male participants between the ages of 18 and 53. A preliminary power analysis for the Wilcoxon test (which is used to conduct the hypotheses, see Section 7) indicates that a sample size of $n = 20$ is sufficient to achieve a power level of 0.9, hence the number of participants. Every participant performed 24 hugs such that each method was deployed on the three non-switching interactions (totaling 12) and on switching interactions for the remaining 12. For every hug, the participant is assigned an interaction and the robot is assigned a method, both of which are randomly drawn without replacement. In the switching interactions, participants performed left-high to right-high hugs, right-high to left-high hugs, and a randomly chosen option from the former two interactions. This setup ensures that both switching and non-switching interactions have the same number of samples for every hypothesis test.

6.5 Quantitative Performance

In addition to the participant study, offline experiments are conducted to quantitatively evaluate the performance of the proposed algorithm against the baseline methods in terms of MSE. All methods predicted a response for the demonstrations in the validation set as well as for demonstrations from a set of unseen switching interactions. To obtain the ground truth dataset for the switching interactions, hug trajectories are manually designed where the robot transitions from left-high to right-high and from right-high to left-high. A total of 25 demonstrations of the left-to-right interaction and 25 demonstrations of the right-to-left interaction are collected. In this case, each of the 5 participants partake in 5 demonstrations per switching interaction, where the robot executes the designed trajectories. After which, outliers are removed in the same manner as the validation set, resulting in 39 demonstrations in total.

Chapter 7

RESULTS AND DISCUSSION

In the following section, the performance of the proposed method is discussed by analyzing participant responses and MSE prediction values with regard to the given hypotheses.

7.1 Survey Responses

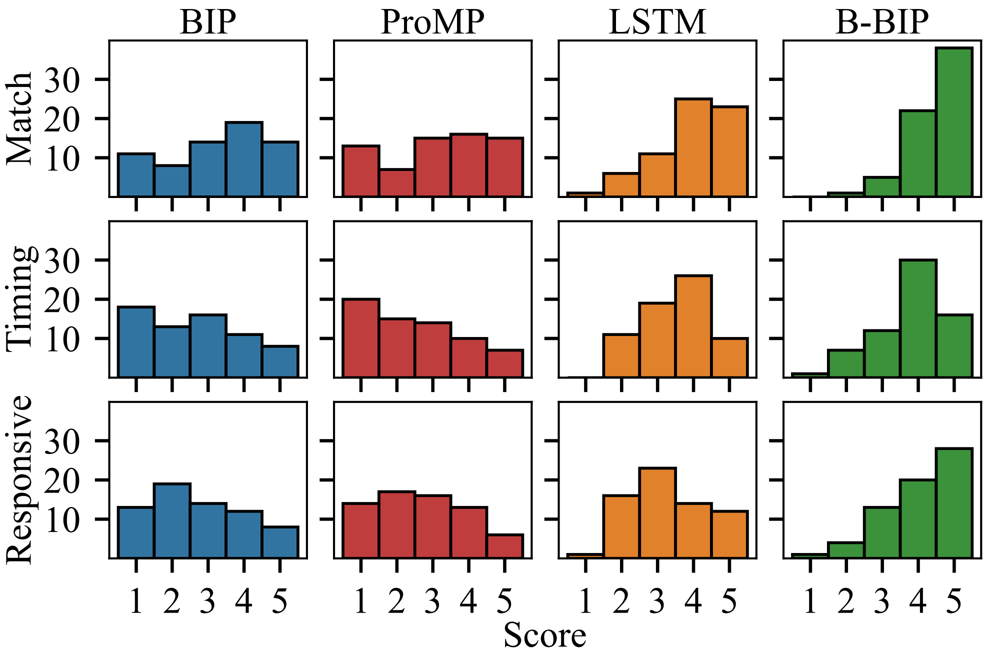

Figure 5. Distribution of scores for the three questions- which are used for hypothesis tests- after switching interactions.

Participant survey responses are visualized in Fig. 5 as a histogram, where a qualitative difference in distributions can be observed. Notably, B-BIP yields the largest number of survey responses with a maximum score of 5 for each question. When performing hypothesis tests, the assumption that responses across different treatments are independent cannot be made; namely, every participant partakes in the same set of methods. Additionally, given that the variable of interest, *score*, takes on an integer value from 1 through 5 (as in the Likert scale) and that the residual of scores differences do not appear to be normally distributed, a paired non-parametric hypothesis test is chosen. A one-way repeated measures analysis of variance is conducted to test for differences in methods across participant responses by using the Friedman test. After obtaining a p-value of $p < 10^{-5}$, post-hoc analysis is performed by applying the two-sided Wilcoxon signed-rank test to every baseline method paired with the proposed method. To account for multiple comparisons, a Bonferroni correction is applied. This procedure is performed separately for non-switching and switching interaction scores. For a given hypothesis test, there are 66 pairs of responses (i.e, 22 participants who perform 3 switching/non-switching interactions), and each pair contains the participant's scores to the question of interest for the methods being compared.

Switching	$\boldsymbol{H_1}$: p-value	$\boldsymbol{H_2}$: p-value	$\boldsymbol{H_3}$: p-value
B-BIP vs. BIP	3.62×10^{-7}	1.94×10^{-6}	6.56×10^{-8}
B-BIP vs. ProMP	6.16×10^{-7}	1.25×10^{-7}	2.27×10^{-7}
B-BIP vs. LSTM	6.64×10^{-4}	9.85×10^{-2}	1.82×10^{-4}
Supported	Yes	Partially	Yes

Non-Switching	$\boldsymbol{H_1}$: p-value	$\boldsymbol{H_2}$: p-value	$\boldsymbol{H_3}$: p-value
B-BIP vs. BIP	1.00	1.21×10^{-1}	1.00
B-BIP vs. ProMP	2.29×10^{-1}	3.08×10^{-3}	1.59×10^{-2}
B-BIP vs. LSTM	4.80×10^{-4}	1.40×10^{-4}	5.70×10^{-5}
Supported	Partially	Partially	Partially

Table 1. Top: p-values for Switching Interactions. Bottom: p-values for Non-Switching Interactions. Grey cells indicate a p-value greater than $\alpha = 0.05$.

The resulting significance results are shown in Table 1. For switching hugs, it was found that participants strongly preferred B-BIP over all baseline methods with respect to hugging type, timing, and responsiveness. Hypotheses $\boldsymbol{H_1}$ and $\boldsymbol{H_3}$ were fully supported with participants reporting B-BIP to offer improved timing and responsiveness over BIP, ProMP, and LSTM. Hypothesis $\boldsymbol{H_2}$ was partially supported, with participants finding B-BIP to yield better timing than BIP and ProMP with statistical significance, but not to LSTM. Responses are more mixed for non-switching hugs, as participants did not prefer B-BIP over BIP in any category, while preferring B-BIP over ProMP only in terms of timing ($\boldsymbol{H_2}$) and responsiveness ($\boldsymbol{H_3}$). These results are not unexpected, as any given single interaction falls within the prior distribution modeled by BIP, which means it is not unreasonable for B-BIP to perform similarly. B-BIP was, however, preferred over the LSTM for all categories, indicating that LSTM struggled to generalize with the small number of training demonstrations.

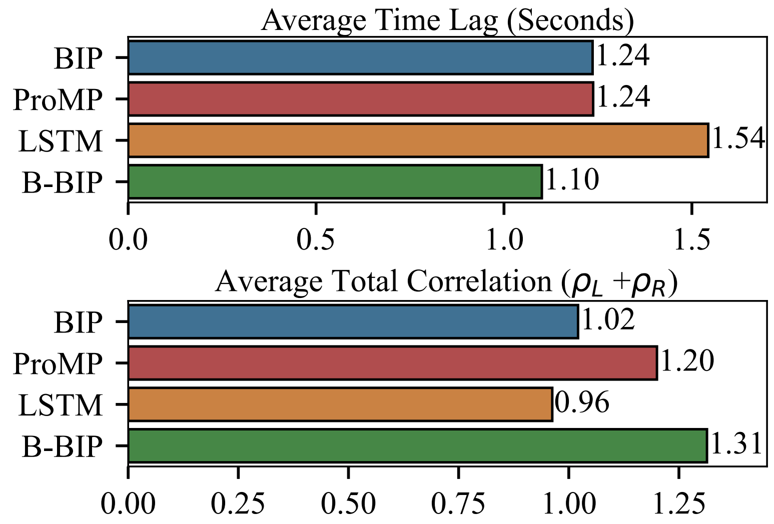

Figure 6. Top: The average of the time lags (in seconds) which maximize the correlation coefficients (below) between human and robot trajectories over all switching interactions from the participant study (see Section 7.2).

In order to further assess the responsiveness, an evaluation of how well the robot matched switching interactions from the group of test participants was performed. The translation (backwards in time, in seconds) was calculated, which, when applied to the robot end effector trajectories, maximizes the sum of the Pearson correlation coefficients from: 1) the z position of the human's right hand with the z position of the robot's eight end effector, and 2) the z position of the human's left hand with the z position of the robot's left end effector. Namely, samples from these matching modalities (from the recorded test interactions) were paired to compute the correlation coefficients, and the average was taken over all switching interactions for every method. These calculations take advantage of the symmetrical nature of the interaction, and the intuition here is that preferable methods should exhibit strong correlations with the participant within a small time lag, especially when the switching occurs- with a perfect response having zero time delay. Figure 6 shows that B-BIP has a correlation which is maximized at the smallest time delay of all methods, and B-BIP produces the highest total correlation (i.e, is the sum of the correlation coefficients from the left and right matching modalities), with a value of 1.31 out of a maximum value of 2.0. These findings provide further support toward the hypothesis that B-BIP exhibits better responsiveness during switching interactions.

7.3 Quantitative Analysis

	Switching	**Non-Switching**
BIP	0.127 ± 0.008	0.040 ± 0.002
ProMP	0.128 ± 0.005	0.036 ± 0.001
LSTM	0.107 ± 0.005	0.102 ± 0.004
B-BIP	0.062 ± 0.005	0.018 ± 0.000

Table 2. MSE values of the controlled DoFs (predicted at each time step) compared to the ground-truth response, using all validation demonstrations. Green cells indicate the method with the smallest mean values. Tukey's Range Test indicates statistical significance for B-BIP in all cases, having $p < 10^{-5}$ in all cases.

The MSE values for all controlled DoFs in the quantitative offline experiments were shown in Table 2. It was found that B-BIP yielded significantly lower prediction errors than all baseline methods for both switching and non-switching hug types. From the significance in switching hug types, it can be concluded that the interaction detection mechanism worked as intended and yielded more accurate inference than BIP. The significance for non-switching hugs was somewhat surprising given the lack of statistical significance in the survey responses for the non-switching hugs in Sec. 7.1, and it it conjectured that error values of this magnitude did not always consistently result in noticeable behavioral differences. It is clear, however, that despite non-switching hugs falling within BIP's prior distribution, the wider distribution resulted in significantly larger spatio-temporal errors when compared to the per-interaction priors of B-BIP.

Chapter 8

CONCLUSION

This thesis presents the mathematical foundations for an imitation learning algorithm capable of adapting to a variety of different hugging interaction scenarios by learning from expert demonstrations. A carefully-designed user study was carried out to validate whether the method produced more responsive, timely, and suitably-matching behavior, all of which were supported through qualitative and quantitative analysis. Notably, Blending BIP achieved 1) statistically significant participant responses on switching interactions for most hypotheses, 2) a nearly twofold reduction in mean-squared prediction error compared to the second-best methods, and 3) the highest correlation and lowest response lag with test participants. These results can hopefully be extended to novel scenarios outside of the scope of this work, especially where variations in human-robot interactions are commonly found.

REFERENCES

Amor, H. B., G. Neumann, S. Kamthe, O. Kroemer and J. Peters, “Interaction primitives for human-robot cooperation tasks”, in “Robotics and Automation (ICRA), 2014 IEEE International Conference on”, pp. 2831–2837 (IEEE, 2014).

Atkeson, C. G. and S. Schaal, “Robot learning from demonstration”, in “ICML”, vol. 97, pp. 12–20 (1997).

Benhamou, E., “Kalman filter demystified: from intuition to probabilistic graphical model to real case in financial markets”, arXiv preprint arXiv:1811.11618 (2018).

Bertsekas, D., *Reinforcement learning and optimal control* (Athena Scientific, 2019).

Block, A. E. and K. J. Kuchenbecker, “Softness, warmth, and responsiveness improve robot hugs”, International Journal of Social Robotics **11**, 1, 49–64 (2019).

Burgers, G., P. J. Van Leeuwen and G. Evensen, “Analysis scheme in the ensemble kalman filter”, Monthly weather review **126**, 6, 1719–1724 (1998).

Campbell, J. and H. Ben Amor, “Bayesian interaction primitives: A slam approach to human-robot interaction”, in “Conference on Robot Learning”, pp. 379–387 (2017).

Campbell, J., A. Hitzmann, S. Stepputtis, S. Ikemoto, K. Hosoda and H. Ben Amor, “Learning interactive behaviors for musculoskeletal robots using bayesian interaction primitives”, in “IEEE/RSJ International Conference on Intelligent Robots and Systems”, (2019a).

Campbell, J., S. Stepputtis and H. Ben Amor, “Probabilistic multimodal modeling for human-robot interaction tasks”, in “Robotics: Science and Systems”, (2019b).

Drolet, M., J. Campbell and H. B. Amor, “Learning and blending robot hugging behaviors in time and space”, IEEE International Conference on Robotics and Automation (ICRA) 2023 (2023).

Evensen, G., “The ensemble kalman filter: Theoretical formulation and practical implementation”, Ocean dynamics **53**, 4, 343–367 (2003).

Ewerton, M., G. Neumann, R. Lioutikov, H. B. Amor, J. Peters and G. Maeda, “Learning multiple collaborative tasks with a mixture of interaction primitives”, in “Robotics and Automation (ICRA), 2015 IEEE International Conference on”, pp. 1535–1542 (IEEE, 2015).

Gelb, A. *et al.*, *Applied optimal estimation* (MIT press, 1974).

Hastie, T., R. Tibshirani, J. H. Friedman and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2 (Springer, 2009).

Hedayati, H., S. Bhaduri, T. Sumner, D. Szafir and M. D. Gross, "Hugbot: a soft robot designed to give human-like hugs", in "Proceedings of the 18th ACM International Conference on Interaction Design and Children", pp. 556–561 (2019).

Ho, J. and S. Ermon, "Generative adversarial imitation learning", Advances in neural information processing systems **29** (2016).

Hochreiter, S. and J. Schmidhuber, "Long short-term memory", Neural computation **9**, 8, 1735–1780 (1997).

Ijspeert, A. J., J. Nakanishi, H. Hoffmann, P. Pastor and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors", Neural computation **25**, 2, 328–373 (2013).

Kalman, R. E., "A new approach to linear filtering and prediction problems", (1960).

Kober, J., J. A. Bagnell and J. Peters, "Reinforcement learning in robotics: A survey", The International Journal of Robotics Research **32**, 11, 1238–1274 (2013).

Maeda, G. J., G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer and J. Peters, "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks", Autonomous Robots **41**, 3, 593–612 (2017).

O'Hagan, A. and J. J. Forster, *Kendall's advanced theory of statistics, volume 2B: Bayesian inference*, vol. 2 (Arnold, 2004).

Paraschos, A., C. Daniel, J. Peters and G. Neumann, "Using probabilistic movement primitives in robotics", Autonomous Robots **42**, 529–551 (2018).

Paraschos, A., C. Daniel, J. R. Peters and G. Neumann, "Probabilistic movement primitives", Advances in neural information processing systems **26** (2013).

Pomerleau, D. A., "Alvinn: An autonomous land vehicle in a neural network", Advances in neural information processing systems **1** (1988).

Ross, S., G. Gordon and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning", in "Proceedings of the fourteenth international conference on artificial intelligence and statistics", pp. 627–635 (JMLR Workshop and Conference Proceedings, 2011).

Schaal, S., "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics", in "Adaptive motion of animals and machines", pp. 261–280 (Springer, 2006).

Shiomi, M., A. Nakata, M. Kanbara and N. Hagita, “A robot that encourages self-disclosure by hug”, in “International Conference on Social Robotics”, pp. 324–333 (Springer, 2017).

Sumioka, H., A. Nakae, R. Kanai and H. Ishiguro, “Huggable communication medium decreases cortisol levels”, Scientific reports **3**, 1, 1–6 (2013).

Sutton, R. S. and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).

Thrun, S., W. Burgard and D. Fox, *Probabilistic robotics* (MIT press, 2005).

Toussaint, M. and C. Goerick, “A bayesian view on motor control and planning.”, From Motor Learning to Interaction Learning in Robots **264**, 227–252 (2010).

Yamazaki, R., L. Christensen, K. Skov, C.-C. Chang, M. F. Damholdt, H. Sumioka, S. Nishio and H. Ishiguro, “Intimacy in phone conversations: anxiety reduction for danish seniors with hugvie”, Frontiers in psychology **7**, 537 (2016).