MILD: Multimodal Interactive Latent Dynamics for Learning Human-Robot Interaction

Vignesh Prasad[1,2], Dorothea Koert[1,5], Ruth Stock-Homburg[2], Jan Peters[1,3,4,5], Georgia Chalvatzaki[1,4]

***Abstract*— Modeling interaction dynamics to generate robot trajectories that enable a robot to adapt and react to a human's actions and intentions is critical for efficient and effective collaborative Human-Robot Interactions (HRI). Learning from Demonstration (LfD) methods from Human-Human Interactions (HHI) have shown promising results, especially when coupled with representation learning techniques. However, such methods for learning HRI either do not scale well to high dimensional data or cannot accurately adapt to changing via-poses of the interacting partner. We propose Multimodal Interactive Latent Dynamics (MILD), a method that couples deep representation learning and probabilistic machine learning to address the problem of two-party physical HRIs. We learn the interaction dynamics from demonstrations, using Hidden Semi-Markov Models (HSMMs) to model the joint distribution of the interacting agents in the latent space of a Variational Autoencoder (VAE). Our experimental evaluations for learning HRI from HHI demonstrations show that MILD effectively captures the multimodality in the latent representations of HRI tasks, allowing us to decode the varying dynamics occurring in such tasks. Compared to related work, MILD generates more accurate trajectories for the controlled agent (robot) when conditioned on the observed agent's (human) trajectory. Notably, MILD can learn directly from camera-based pose estimations to generate trajectories, which we then map to a humanoid robot without the need for any additional training. Supplementary Material: `https://bit.ly/MILD-HRI`**

I. Introduction

Observing human actions and interacting synchronously is an essential characteristic of a social robot in HRI scenarios [1]. Key components for learning coordinated HRI policies are having a good spatio-temporal representation and jointly modeling the interaction dynamics of the agents. In this regard, the paradigm of LfD shows promising results [2], [8], [7], especially when using only a handful of trajectories. Such LfD approaches learn joint distributions over human and robot trajectories that can be conditioned on the observed actions of the human, although, they scale poorly with higher dimensions. In such cases, deep LfD approaches perform well for learning latent-space dynamics with high-dimensional data [20], [11], [9], [13], [29] but they usually are not scalable to different interactive scenarios, as they usually do not model the inherent multimodality and uncertainty of HRI tasks.

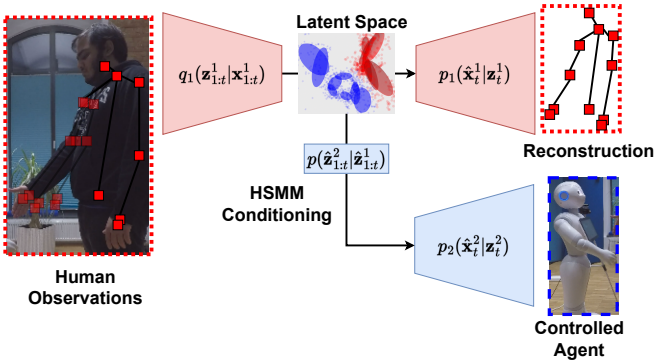


Fig. 1: Overview of our approach, "MILD". We train VAEs to reconstruct the observations of the interactions agents $(\boldsymbol{x}^1_{1:t}, \boldsymbol{x}^2_{1:t})$ with an HSMM prior to learn a joint distribution over the latent space trajectories $(\boldsymbol{z}^1_{1:t}, \boldsymbol{z}^2_{1:t})$ of the interacting agents. During test time, the observed agent's latent trajectory conditions the HSMM to infer the controlled agent's latent trajectory $p(\boldsymbol{z}^2_t|\boldsymbol{z}^1_{1:t})$ which is decoded to generate the agent's real-world trajectory $\hat{\boldsymbol{x}}^2_t$.

To tackle these challenges when learning HRI policies, we introduce MILD, a method that effectively couples benefits from deep LfD methods with probabilistic machine learning. MILD uses HSMMs as a temporally coherent latent space prior of a VAE, which we learn from HHI data. Specifically, we model the prior as a joint distribution over the trajectories of both interacting agents, making full use of the power of HSMMs for learning both trajectory and interaction dynamics. Our approach successfully captures the multimodality of the latent interactive trajectories thanks to the modularity of HSMMs, enabling better modeling of dynamics as compared to using an uninformed, stationary prior [5]. During testing, we can generate latent trajectories by conditioning the HSMM on the human observations using Gaussian Mixture Regression [7], and decode them to obtain the robot's control trajectories (Fig. 1).

Our experimental evaluations on different test scenarios show the efficacy of MILD in capturing coherent latent dynamics, both in predicting HHI and in generating controls for HRI on different robots, compared to the state-of-the-art method that implicitly learns shared representations [5]. MILD learns to generate effective robot trajectories for HRI,

[1] Department of Computer Science, TU Darmstadt, Germany.
[2] Chair for Marketing and Human Resource Management, Department of Law and Economics, TU Darmstadt, Germany.
[3] German Research Center for AI (DFKI), Research Department: Systems AI for Robot Learning.
[4] Hessian.AI
[5] Centre for Cognitive Science, TU Darmstadt, Germany.
This work was supported by the German Research Foundation (DFG) Project "Social Robots at the Customer Interface" (Grant No.: STO 477/14-1), the DFG Emmy Noether Programme (CH 2676/1-1), the German Federal Ministry of Education and Research (BMBF) Project "IKIDA" (Grant no.: 01IS20045), the RoboTrust project of the Centre Responsible Digitality (ZEVEDI) Hessen, Germany, the Funding Association for Market-Oriented Management, Marketing, and Human Resource Management (Förderverein für Marktorientierte Unternehmensführung, Marketing und Personal management e.V.), and the Leap in Time Foundation (Leap in Time Stiftung). The authors thank the NHR Centers NHR4CES at TU Darmstadt for the access to the Lichtenberg high-performance computer (Project No. 1694) for running the experiments in this work.

not only through robot kinesthetic teaching, but also by learning from HHI, both from idealistic data (Motion Capture) and noisy RGB-D skeleton tracking, by directly transferring the generated trajectories to a humanoid robot [17], [33], without requiring additional demonstrations or fine-tuning.

II. Related Work

Early approaches for learning HRI modeled them as a joint distribution with a Gaussian Mixture Model (GMM) learned over demonstrated trajectories of a human and a robot in a collaborative task [7]. The correlations between the human and the robot degrees of freedom (DoFs) can then be leveraged to generate the robot's trajectory given observations of the human. This method was further extended with HSMMs with explicit duration constraints for learning both proactive and reactive controllers [35]. Along similar lines of leveraging Gaussian approximations for LfD, Movement primitives [31], [36], which learn a distribution over underlying weight vectors obtained via linear regression, were extended for HRI by similarly learning a joint distribution over the weights of both interacting agents [2], [26]. The versatility of interaction primitives can additionally be noted by their ability to be adapted for different intention predictions [22], speeds [27], or for learning multiple interactive tasks seamlessly by either using a GMM as the underlying distribution [16] or in an incremental manner [28], [23].

Deep LfD techniques have grown in popularity for learning latent trajectory dynamics from demonstrations wherein an autoencoding approach, like VAEs, is used to encode latent trajectories over which a latent dynamics model is trained. In their simplest form, the latent dynamics can be modeled either with linear Gaussian models [20] or Kalman filter [3]. Other approaches learn stable dynamical systems, like Dynamic Movement Primitives [36] over VAE latent spaces [4], [10], [11], [9]. Instead of learning a feedforward dynamics model, Dermy et al. [13] model the entire trajectory's dynamics at once using Probabilistic Movement Primitives [31] achieving better results than [9]. When large datasets are available, Recurrent Networks are powerful tools in approximating latent dynamics [12], [18], especially in the case of learning common dynamics models in HRIs [5]. A major advantage of most of the aforementioned LfD approaches, other than their sample efficiency in terms of demonstrations, is that they can be explicitly conditioned at desired time steps, unlike neural network-based approaches.

Most deep LfD approaches fit complete trajectories, curating neither the multimodality in HRI tasks nor the subsequent dynamics between different modes of interaction. Instead of fitting a single distribution over demonstrated trajectories, HSMMs break down such complex trajectories into multiple modes and learn the sequencing between hidden states, as shown in [29], where HSMMs were used as latent priors for a VAE. However, [29] does not look at the interdependence between dimensions, but models each dimension individually, which is not favorable when learning interaction dynamics. Such issues can be circumvented by using a diagonal cross-covariance structure (as in [3]), but this would only learn dependencies between individual dimensions. Contrarily, we learn full covariance matrices in our HSMM models to better facilitate the learning of interaction dynamics.

III. Preliminaries

In this section, we briefly introduce preliminary concepts, namely, VAEs (Sec. III-A) and HSMMs (Sec. III-B), that we deem useful for discussing our proposed method.

A. Variational Autoencoders

Variational Autoencoders (VAEs) [21], [34] are a style of neural network architectures that learn the identity function in an unsupervised, probabilistic way. An encoder encodes the inputs onto a latent space, denoted by "$\boldsymbol{z}$", of the input "$\boldsymbol{x}$" at the bottleneck that a decoder uses to reconstruct the original input. A prior distribution is enforced over the latent space, usually given by a normal distribution $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \boldsymbol{I})$. The goal is to estimate the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$, using a neural network $q(\boldsymbol{z}|\boldsymbol{x})$ and is trained by minimizing the Kullback-Leibler (KL) divergence between them.

$$D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})) = \mathbb{E}_q[\log \frac{q(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{x}, \boldsymbol{z})}] + \log p(\boldsymbol{x}) \quad (1)$$

This can be re-written as

$$\log p(\boldsymbol{x}) = D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})) + \mathbb{E}_q[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}] \quad (2)$$

The KL divergence is always non-negative, therefore the second term in (2) acts as a lower bound. Maximizing it would effectively maximize the log-likelihood of the data distribution or evidence, and is hence called the Evidence Lower Bound (ELBO), which can be written as

$$\mathbb{E}_q[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}] = \mathbb{E}_q[\log p(\boldsymbol{x}|\boldsymbol{z})] + D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \quad (3)$$

The first term corresponds to the reconstruction of the input via samples decoded from the posterior distribution. The second term is the KL divergence between the prior and the approximate posterior, which acts as a regularization term for the posterior. Further information about variational inference can be found in [21], [34].

B. Hidden Semi-Markov Models

HSMMs are a special class of Hidden Markov Models (HMMs), where the Markov property is relaxed, i.e., the current state depends on not just the previous state, but also the duration for which the system remains in a state. In an HMM, a sequence of observations $\boldsymbol{z}_{1:T}$ is modeled as a sequence of K hidden latent states that emit the observations with some probability. Specifically, an HMM can be described by its initial state distribution π_i over the states $i \in \{1, 2 \dots K\}$, the state transition probabilities $\mathcal{T}_{i,j}$ denoting the probability of transitioning from state i to state j. In our case, each state is characterized by a Normal distribution with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$, which characterize the emission probabilities of the observations $\mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. This, in essence, is similar to learning a GMM over the observations and learning the

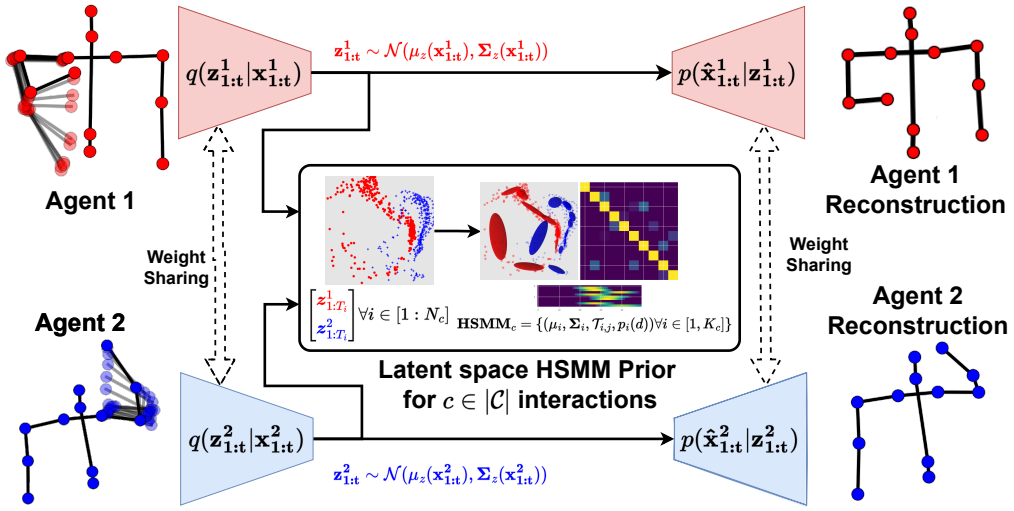


Fig. 2: Pipeline for training MILD: We model the interaction dynamics between two agents in the latent space of a VAE using a temporally coherent prior in the form of an HSMM to model a joint distribution over the latent trajectories. To match the predictive abilities of the HSMM prior, we predict full posterior covariance matrices, rather than forcing a diagonalized covariance like traditional VAEs. The use of an HSMM prior, as opposed to standard Gaussians or other LfD approaches, enforces a modularized latent space using a GMM $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ which subsequently learns the transitions between clusters $\mathcal{T}_{i,j}$ as well as duration statistics $p_i(d)$ to improve the transition prediction.

temporal sequencing between the Gaussian components, by learning the forward variable of the HMM $h_i(\boldsymbol{z}_t)$

$$h_i(\boldsymbol{z}_t) = \alpha_i(\boldsymbol{z}_t) / \sum_{k=1}^{K} \alpha_k(\boldsymbol{z}_t) \tag{4}$$

where

$$\alpha_i(\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \sum_{k=1}^{K} \alpha_k(\boldsymbol{z}_{t-1}) \mathcal{T}_{k,i} \tag{5}$$

and $\alpha_i(\boldsymbol{z}_0) = \pi_i$. In the case of HSMMs, a distribution $p_i(d)$ is additionally fitted over the number of steps $d \in \{1, 2, \dots D\}$ that the model stays in a given state

$$\alpha_i(\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \sum_{k=1}^{K} \sum_{d=1}^{D} \alpha_k(\boldsymbol{z}_{t-1}) \mathcal{T}_{k,i} p_i(d) \tag{6}$$

For a more in-depth explanation of training GMMs, HMMs and HSMMs in the context of robot learning, we refer the reader to [6], [32]. To encode the joint distribution between the interacting agents, an observation space is constructed by concatenating the DoFs of both agents [7], [15]. This allows the distribution to be decomposed as

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^1 \\ \boldsymbol{\mu}_i^2 \end{bmatrix}; \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^{11} & \boldsymbol{\Sigma}_i^{12} \\ \boldsymbol{\Sigma}_i^{21} & \boldsymbol{\Sigma}_i^{22} \end{bmatrix} \tag{7}$$

Once the distributions are learned, given observations $\boldsymbol{z}_{1:t}^1$ of Agent 1, the trajectory for Agent 2 $\boldsymbol{z}_{1:t}^2$ is inferred as

$$\boldsymbol{z}_{1:t}^2 = \sum_{k=1}^{K} h_k(\boldsymbol{z}_{1:t}^1)(\boldsymbol{\mu}_k^2 + \boldsymbol{\Sigma}_k^{21}(\boldsymbol{\Sigma}_k^{11})^{-1}(\boldsymbol{\mu}_i^1 - \boldsymbol{z}_{1:t}^1)). \tag{8}$$

This allows the HSMM to adapt the controlled agent's trajectory to the observed agent thereby capturing the interaction dynamics and resulting in more accurate interaction.

IV. Multimodal Interactive Latent Dynamics

In this section, we introduce our approach, MILD (Multimodal Interactive Latent Dynamics), which uses the representation learning abilities of VAEs coupled with the abilities of HSMMs for interaction modeling by incorporating the HSMMs as the prior in the VAE. Our method extends the idea proposed in [29] who showed promising results for trajectory segmentation using an HSMM prior with a VAE. However, in [29], the correlation between different dimensions is not modeled, which is a key factor when jointly modeling the interaction dynamics in HRI scenarios. Therefore, rather than learning a single model per dimension, we model full covariance matrices in the HSMM, allowing us to predict the inter-dependency between human and robot DOFs. We describe the approach of training a VAE with an HSMM prior in Sec. IV-A, which is visualized in Fig. 2, followed by how the robot actions can be interactively generated by conditioning the HSMM in Sec. IV-B.

A. Learning VAEs with HSMMs priors for HRI

Typically in VAEs, the prior $p(\boldsymbol{z})$ is modeled as a stationary distribution that doesn't change with time. When it comes to learning trajectories, this can be a very hard constraint. We argue that having meaningful transition priors can help learn temporally coherent latent spaces [11]. Modeling trajectory priors could potentially be achieved by autoencoding subsequences rather than individual time-steps like in [5], but it still requires learning a temporal dynamics model to represent the progression of the trajectory. To this end, we explore the use of HSMMs that not only learn latent-space dynamics but are also able to break down the interactions into multiple phases and learn their sequencing, leading to a more modular structure, and can also learn inter-dependencies between actions. We defer this direction of jointly learning action segmentation and the underlying modular skills to

future work, and we mainly focus on learning latent-space HSMMs for HRI to capture the task-wise multimodality.

In order to capture the interactive behavior, we learn a joint distribution over the trajectories of both interacting agents. Given a latent encoding of the trajectories of both interacting agents $\boldsymbol{z}^1_{1:T}$ and $\boldsymbol{z}^2_{1:T}$, the prior probabilities of the encoded observations at each time-step $\boldsymbol{z}_t$ of an input $\boldsymbol{x}_t$ are calculated using the current most likely component of the HSMM $\hat{i}$ (without any observation, by dropping the emission probability from (4) and (6)) whose parameters are used as the prior in the ELBO.

$$\begin{aligned} \hat{i} &= \arg\max_i h_i(\cdot) \\ p(\boldsymbol{z}^s_t) &= \mathcal{N}(\boldsymbol{z}^s_t; \boldsymbol{\mu}^s_{\hat{i}}, \boldsymbol{\Sigma}^{ss}_{\hat{i}}); s \in \{1,2\} \end{aligned} \tag{9}$$

where s denotes each agent's index. The HSMM parameters of all the interactions are fixed during a training epoch. After the training of the VAE for an epoch, the new HSMM parameters are estimated using the learned trajectory encodings.

VAEs model the posterior distribution with a simple diagonalized covariance matrix, whereas the HSMM distributions are modeled with a full covariance matrix that captures the relations between the different DoFs. Therefore, we predict the full covariance matrix as well, to better represent the correlation between dimensions of the demonstrated trajectories. However, this is not a straightforward task, as we need to ensure that the predicted matrix is symmetric and positive definite. While this can be approximated by finding the closest symmetric positive definite matrix within its Frobenius norm [19], it involves computing the eigen-decomposition of the matrix and reconstructing it back for each data point. This makes it computationally expensive, especially when it involves backpropagation through the networks for a large number of samples. Therefore, we follow the approach presented in [14] by predicting the lower triangular matrix $\boldsymbol{L}$ corresponding to the Cholesky decomposition of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$. The covariance matrix can be made positive definite by enforcing the diagonal elements of $\boldsymbol{L}$ to be positive. Our posterior can now be written as $q(\boldsymbol{z}_t|\boldsymbol{x}_t) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}_t), \boldsymbol{\Sigma}(\boldsymbol{x}_t))$, where $\boldsymbol{\mu}(\boldsymbol{x}_t)$ and $\boldsymbol{\Sigma}(\boldsymbol{x}_t) = \boldsymbol{L}(\boldsymbol{x}_t)\boldsymbol{L}^T(\boldsymbol{x}_t)$ are the outputs of a neural network. This allows us to model a full covariance matrix that can match the form of the prior.

Alg. 1 provides an overview of the training of MILD. Given a set of labelled trajectories $\boldsymbol{X}$ containing demonstrations of two agents $\boldsymbol{X}^1_{1:T}$ and $\boldsymbol{X}^2_{1:T}$ for $\mathcal{C}$ interactions, in each epoch, the VAE is trained by optimizing (3) using the prior according to (9). Subsequently, an HSMM corresponding to each of the $c \in |\mathcal{C}|$ interactions is trained with the demonstrations of that interaction of both agents jointly using Expectation Maximization as explained in [32].

B. Conditional Trajectory Generation

During testing, given t observations of the human agent $\boldsymbol{x}^1_{1:t}$, we first encode the observations $\boldsymbol{z}^1_{1:t} \sim q_1(\cdot|\boldsymbol{x}^1_{1:t})$ and then condition the learned joint distribution of the given action using the latent encodings of the observed trajectory.

Algorithm 1: Training MILD

Data: A set of trajectories with action labels $\boldsymbol{X} = \{\boldsymbol{X}^1_{1:T}, \boldsymbol{X}^2_{1:T}, c\}$ for $|\mathcal{C}|$ actions
Result: VAE weights and $|\mathcal{C}|$ HSMM parameters

Initialize VAE weights randomly;
Initialize $\boldsymbol{\mu}^c_i \leftarrow \boldsymbol{0}, \boldsymbol{\Sigma}^c_i \leftarrow \boldsymbol{I}\ \forall c \in [1, |\mathcal{C}|]\ \forall i \in [1, K_c]$
while *not converged* **do**
  for $\boldsymbol{x}^1_{1:T}, \boldsymbol{x}^2_{1:T}, c \in \boldsymbol{X}$ **do**
    for $s \in \{1,2\}$ Maximize $\mathbb{E}_q[\log p_s(\boldsymbol{x}^s_t|\boldsymbol{z}^s_t)] + D_{KL}(q_s(\boldsymbol{z}^s_t|\boldsymbol{x}^s_t)||p_c(\boldsymbol{z}^s_t))$;
    where $p_c(\boldsymbol{z}^s_t)$ is calculated using Eq. 9
  end
  for $c \in [1, |\mathcal{C}|]$ **do**
    $\boldsymbol{X}^c \leftarrow$ set of demonstrations of Interaction c;
    $\boldsymbol{Z}^c \leftarrow \emptyset$;
    for $\boldsymbol{x}^1_{1:T}, \boldsymbol{x}^2_{1:T}, c \in \boldsymbol{X}^c$ **do**
      $\boldsymbol{z}^1_{1:T} \sim q_1(\cdot|\boldsymbol{x}^1_{1:T}); \boldsymbol{z}^2_{1:T} \sim q_2(\cdot|\boldsymbol{x}^2_{1:T})$
      $\boldsymbol{Z}^c \leftarrow \boldsymbol{Z}^c \cup \begin{bmatrix} \boldsymbol{z}^1_{1:T} \\ \boldsymbol{z}^2_{1:T} \end{bmatrix}$;
    end
    Train the c^{th} HSMM with $\boldsymbol{Z}^c$
  end
end

The conditioned HSMM model is then used to generate the latent trajectory of the second agent $\hat{\boldsymbol{z}}^2_{1:t}$ using (8), which is then decoded to obtain the actions of the second agent $\hat{\boldsymbol{x}}^2_{1:t} \sim p_2(\cdot|\boldsymbol{z}^2_{1:t})$. This process is shown in Alg. 2. We currently do not perform action recognition to select which HSMM to condition, which we defer to future work.

Algorithm 2: Conditioning on Human Observations

Data: An observation of the human agent $\boldsymbol{x}^1_{1:t}$, Trained MILD Model
Result: Conditioned Trajectory for the second agent $\hat{\boldsymbol{x}}^2_{1:t}$

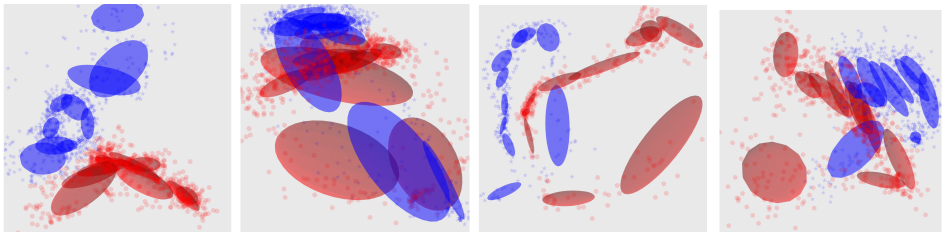
Encode the observed trajectory $\boldsymbol{z}^1_{1:t} \sim q_1(\cdot|\boldsymbol{x}^1_{1:t})$
Condition the HSMM model to get the latent trajectory for the second agent using (8)
$\hat{\boldsymbol{z}}^2_{1:t} = \sum_{k=1}^{K} h_i(\boldsymbol{z}^1_{1:t})(\boldsymbol{\mu}^2_i + \boldsymbol{\Sigma}^{21}_i(\boldsymbol{\Sigma}^{11}_i)^{-1}(\boldsymbol{\mu}^1_i - \boldsymbol{z}^1_{1:t}))$
Decode the conditioned trajectory $\hat{\boldsymbol{x}}^2_{1:t} \sim p_2(\cdot|\hat{\boldsymbol{z}}^2_{1:t})$

V. Experiments and Results

In this section, we first explain our setup's implementation (Sec. V-A) and the datasets used (Sec. V-B), following which we present some qualitative results of the learned latent spaces (Sec. V-C). Finally, we present the results of predicting the controlled agent's trajectories after conditioning on the observed agent in Sec. V-D.

A. Experimental Setup

The networks are implemented using PyTorch with a similar structure as in [5] Both the encoder and the decoder consist of 2 feedforward hidden layers with dimensions [250, 150] with Leaky ReLU activations and 5-dimensional latent space. The encoder has two output networks, one for the mean $\boldsymbol{\mu}_{\boldsymbol{z}}$ and one for the lower triangular matrix $\boldsymbol{L}_{\boldsymbol{z}}$ corresponding to the Cholesky decomposition of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{z}} = \boldsymbol{L}_{\boldsymbol{z}}\boldsymbol{L}^T_{\boldsymbol{z}}$. The diagonal elements



(a) Handshaking (b) Hand Waving (c) Rocket Fistbump (d) Parachute Fistbump

Fig. 3: 3D Latent spaces learned by MILD on the different interactions in [5]. (blue stars - Agent 1, red circles - Agent 2)

are forced to be positive as $l_{\boldsymbol{z}}^{ii} = 2|l_{\boldsymbol{z}}^{ii}|$. To stabilize the VAE training and prevent issues from over regularization, we multiply the KL loss with a scale factor of 10^{-3}. We additionally decode 10 samples from the posterior to train the networks rather than a single sample, which we found helps improve the training. The networks are trained with a learning rate of 10^{-4} using the AdamW optimizer. The HSMMs are implemented using PbDLib[1] [32] with 10 GMM components (chosen empirically) for each HSMM. The HSMMs are initialized by splitting the data temporally into equal components.

B. Dataset

1) Bütepage et al. [5]: [2] capture their HHI data using a Rokoko Smart suit for the human skeleton data and use kinesthetic teaching on an ABB YuMi-IRB 14000 robot for collecting Human-Robot (HRI) data. It consists of demonstrations of 4 actions: Hand Waving, Handshaking, and two different kinds of fist bumps. The first is called Rocket Fistbump, which involves bumping the fists at a low level, followed by raising the fists upwards while in contact with each other. The second is called Parachute Fistbump in which partners bump their fists at a high level near the head and bring it down with simultaneously oscillating the hands sideways, while in contact with each other. A more detailed explanation of the interactions can be found in [5].

In total, there are 149 trajectories for training and 32 for testing from the HHI setting and 32 training and 9 testing for the HRI setting, as in [5]. The trajectories are pre-processed to use the 3D coordinates of 4 right arm joints (as seen in Fig. 5 and 6), with the origin at the shoulder. Like in [5], we concatenate 40 observations corresponding to a time window of 1 second, leading to an input size of 480 dimensions (40x4x3). For the Robot trajectories, the data is similarly sampled, leading to an input size of 280 dimensions (40x7) for the 7 joint angles of the robot's right arm. The hidden layers of the Robot VAE are initialized using the weights from the Human VAE to accelerate the learning given the lower number of robot trajectory samples.

2) Nuitrack Skeleton Interaction Dataset: (NuiSI) is a dataset that we collected ourselves, which consists of 6 interactions, namely Handshaking, Hand waving, High Fives, Fist bumps, Clap fist (Handclap followed by fist bump), and Rocket Fist bump (like in [5]). The data is recorded using an Intel Realsense D435 which provides RGB-D images using Nuitrack[3] for tracking the upper body skeleton joints in each frame. The data is first cleaned to remove any missing trajectories, and the trajectories of both agents are manually aligned and down-sampled to match the trajectory of shorter length. For training, the data is processed similarly as mentioned above with a window size of 5 time steps given the lower frame rate and irregularities during skeleton tracking. We use the pre-trained network trained on the aforementioned skeleton data as well, to help accelerate the learning. We additionally evaluate MILD on the humanoid robot Pepper [30] after training it on the Nuitrack data by making use of the similar DoFs between a human and a humanoid [17], [33] to extract the joint angles for the robot. We use a moving average filter to reduce the jerk of the generated robot trajectories.

C. Learned Representations

Fig. 3 shows some examples of the latent space distributions learned by MILD on the HHI data of [5]. It can be seen that MILD is able to capture most of the different modes of the distribution, rather than forcing them to fit a standard normal distribution. Despite the relatively low spread of the data, the HSMM prior nicely encapsulates the shape of the trajectories, and learns the corresponding sequencing between the modes, thereby effectively capturing the dynamics of the trajectories, as in [29]. However, given the full structure of our covariance matrices, MILD is able to capture the interactivity between the agents as well.

Method		MILD	[5]
Prediction MSE for Agent 2 (cm)	Wave	**0.669 ± 0.910***	1.097 ± 1.719
	Shake	**0.360 ± 0.301***	0.931 ± 0.728
	Rocket	**0.391 ± 0.586***	1.295 ± 1.344
	Parachute	**0.610 ± 0.100***	1.101 ± 0.668

TABLE I: Prediction MSE (in cm) of MILD compared with [5] on the dataset in [5] when predicting the trajectory of the second agent after observing the first agent. (Lower is better, * - $p < 0.05$)

D. Conditioned Prediction Results

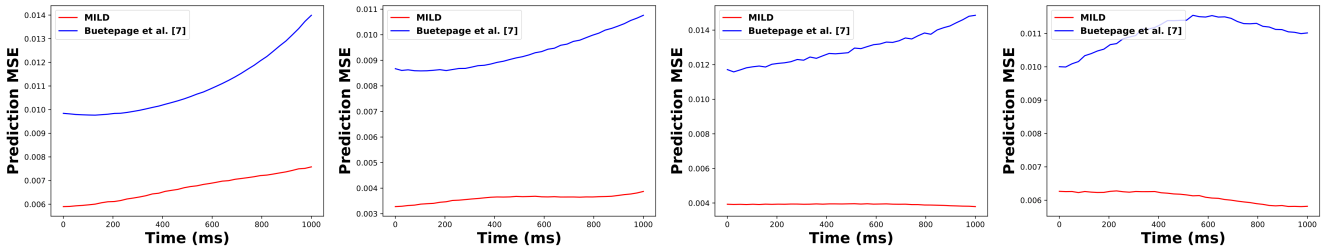
We test the conditioning ability of MILD compared to [5][4] to evaluate the accuracy of the generated motions of the controlled agent after observing the human interaction-partner. We evaluate the approaches over the four interactions of

[1] https://gitlab.idiap.ch/rli/pbdlib-python

[2] https://github.com/jbutepage/human_robot_interaction_data

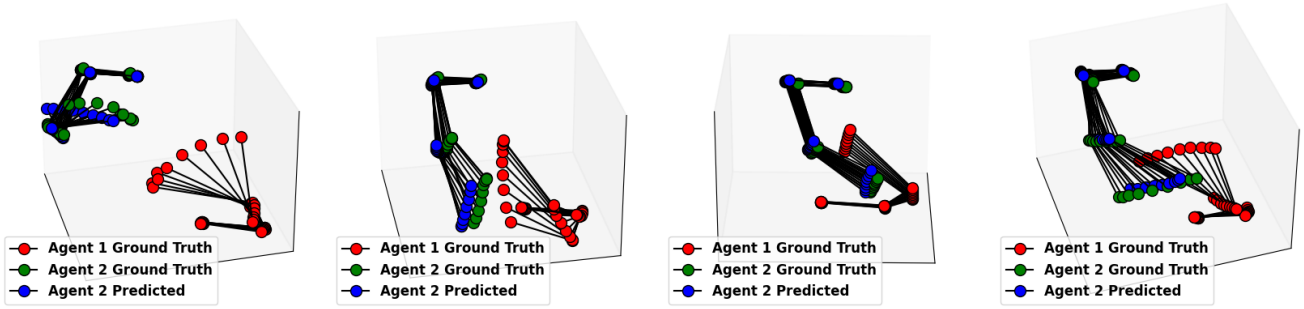
[3] https://nuitrack.com/

[4] Results reported using our own implementation of [5].



(a) Hand Waving (b) Handshaking (c) Rocket Fistbump (d) Parachute Fistbump

Fig. 4: Prediction MSE of the second agent after observing the first agent averaged over 1 second (40 observations) of MILD compared with [5]. (Lower is better, Red - MILD, Blue - [5])



(a) Hand Waving (b) Handshaking (c) Rocket Fistbump (d) Parachute Fistbump

Fig. 5: Sample Human-Human Interactions generated by MILD from the dataset in [5]. The 3D plots consist of 4 right arm joints of the interacting agents, along with the predictions of MILD of the second agent's joints after having observed the first agent. The points, both observed and predicted, correspond to a time window of 1 second (40 observations of which we visualize 8 for ease of vieweing). (Red - Ground Truth Trajectory of Agent 1, Green - Ground Truth Trajectory of Agent 2, Blue - Predicted Trajectory of Agent 2)

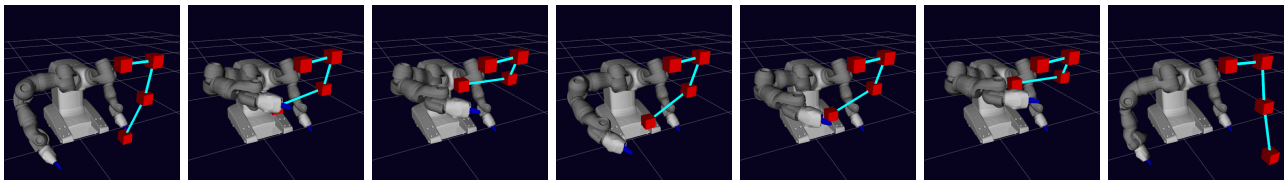


Fig. 6: Example of a Handshaking HRI generated by MILD on the Yumi robot after training on [5]. The Red boxes are the observed right arm 3D coordinates of the human partner.

Method		MILD	[5]
Prediction MSE for the Yumi Robot (rad)	Wave	**0.084** $\pm$ **0.089***	0.348 $\pm$ 0.363
	Shake	**0.038** $\pm$ **0.035***	0.371 $\pm$ 0.224
	Rocket	**0.077** $\pm$ **0.056***	0.362 $\pm$ 0.398
	Parachute	**0.067** $\pm$ **0.049***	0.491 $\pm$ 0.224

TABLE II: Prediction MSE (in radians) for comparing our method, MILD, with [5] on the HRI trajectories from [5]. We predict the trajectory of the robot after observing the human partner. (Lower is better, * - $p < 0.05$)

Action	Prediction MSE (cm)
Clap Fist	0.315 $\pm$ 0.173
Fist Bump	0.195 $\pm$ 0.137
Handshake	0.173 $\pm$ 0.109
High Five	0.293 $\pm$ 0.191
Rocket Fistbump	0.422 $\pm$ 0.495
Hand Wave	1.325 $\pm$ 1.219

TABLE III: Prediction MSE (in cm) of MILD on the NuiSI dataset when predicting the trajectory of the second agent after observing the first agent. (Lower is better)

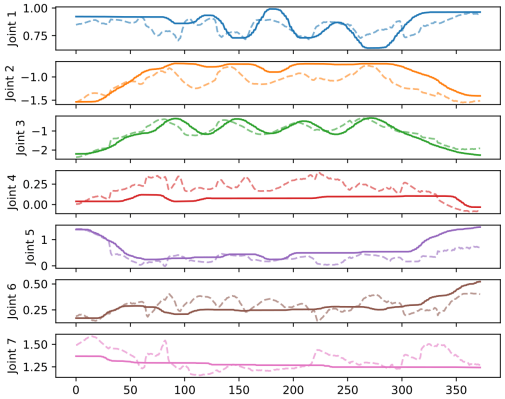
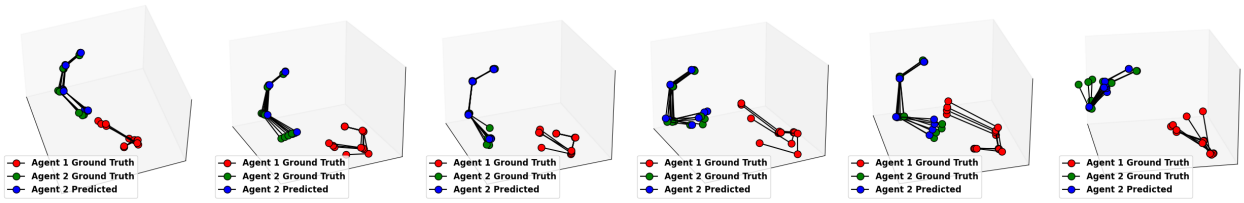


Fig. 7: An example of a trajectory generated by MILD for a Handshaking HRI on the Yumi robot. The dotted lines show the predicted robot joint trajectories and the solid lines show the ground truth (X axis - Time steps, Y axis - Joint angles (radians)).

the dataset in [5]. We calculate the mean squared error (MSE) between the predicted motions and the ground truth



(a) Clap Fist (b) Fist Bump (c) Handshake (d) High Five (e) Rocket Fistbump (f) Hand Wave

Fig. 8: Sample Human-Human Interactions generated by MILD from the NuiSI dataset. The 3D plots consist of 4 right arm joints of the interacting agents, along with the predictions of MILD of the second agent's joints after having observed the first agent. We show the observed and predicted values over 5 time steps. (Red - Ground Truth Trajectory of Agent 1, Green - Ground Truth Trajectory of Agent 2, Blue - Predicted Trajectory of Agent 2)

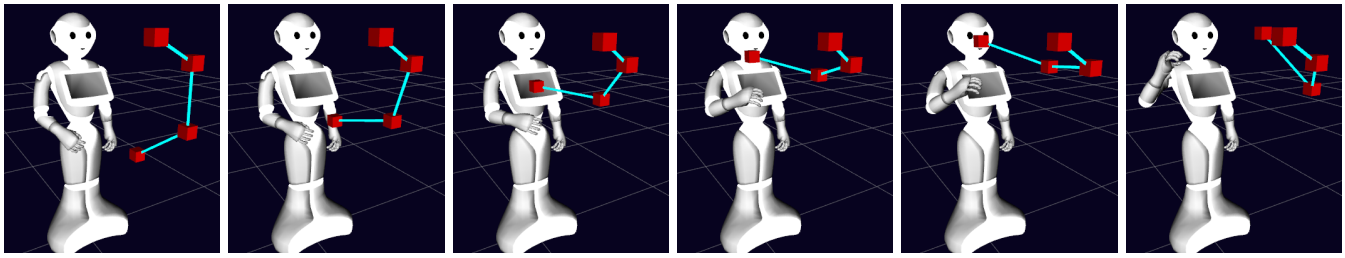


Fig. 9: Example of a Rocket Fistbump HRI generated by MILD on the Pepper robot after training on the NuiSI dataset. The Red boxes are the observed right arm 3D coordinates of the human partner.

skeleton/robot joints at each time step after observing the human's trajectory. The results of these interactions can be seen in Fig. 4 and Table I. We observe that MILD consistently performs better than [5] and accurately predicts the motions of the interacting agent over all the different interactions. One specific reason behind this could be that in [5], they learn a common spatial latent space followed by a dynamics latent space, regularizing the spatial embeddings of each agent to match temporally. During testing, this could create errors when generating from the temporal latent space, since there is no way to distinguish whether a given sample corresponds to the first agent or the second agent. This shows the power of the explicit conditioning ability of HSMMs in predicting interactive behaviors since the agents can be jointly modeled, as compared to learning just a single representation over the two agents. Some qualitative results of the predicted skeletons from the dataset in [5] is can be seen in Fig. 5, where the trajectories of both the interacting agents over a 1 second period (8 out of 40 time steps) and the predictions of MILD for the second agent are visualized.

We additionally show the results of training on the HRI trajectories from the dataset in [5]. In this case, the accuracy obtained using MILD (Table. II) outperforms [5] over all the interactions, showing that our method is able to effectively model not just HHI tasks but also HRI tasks. However, the general accuracy is not as good as the HHI scenario, as can be seen in Fig. 7. This can be attributed to the relatively small size of the dataset (only 32 HRI trajectories are present in the dataset, compared to 149 HHI trajectories). In such cases, learning HRI from HHI can yield a more viable solution. Despite this, we are able to nicely capture the temporal dependencies between the human and the robot, which can be seen in the interactivity of an example Human-Robot handshaking exchange in Fig. 6. Here, we show how an HRI between a simulated Yumi robot and the observed (simulated) human, whose right arm joints are shown in red. It can be seen that the motions of the robot are in sync (within the 1 second delay) with that of the human during the handshake, which shows that MILD is able to model the interactivity in such tasks.

On the NuiSI dataset, MILD is able to achieve good accuracy (within cm range as shown in Table III) in predicting the skeleton trajectories of the second (controlled) agent after observing the first agent. Examples of these trajectories can be seen in Fig. 8 where one can see the accuracy of our predicted skeletons (in blue). This shows that MILD performs well not just on idealistic motion capture data, but also on real-world interaction scenarios involving relatively noisy inputs. We additionally show some quantitative results of MILD on NuiSI by mapping the learned trajectories to a Pepper Robot. We do so by making use of the similarities in the DoFs between a human arm and Pepper, using which the joint angles are calculated from the generated skeletons, which are then executed on the robot [17], [33]. An example of a Rocket Fistbump HRI with a simulated Pepper can be seen in Fig. 9. Please refer to our supplementary video and site for the real-robot demonstration.

VI. Conclusion and Future Work

In this paper, we proposed an extension to the general idea of using Learning for Demonstration (LfD) techniques for learning latent space dynamics from human observations. We proposed Multimodal Interactive Latent Dynamics (MILD), a method that combines Hidden Semi-Markov Models (HSMMs) as prior in Variational Autoencoders (VAEs) for learning Human-Robot Interaction. The predictive ability of HSMMs, allows us to effectively model a joint distribution

over the demonstrated trajectories from Human-Human interactions. We can generate robot control trajectories during test time by conditioning HSMMs on the human-partner actions in a timely and reactive manner. Our approach performs better than the current state-of-the-art for learning latent interaction dynamics in HRI, which we show through extensive experiments on a diverse range of datasets.

Currently, we do not incorporate any action recognition for selecting the corresponding latent HSMM that needs to be executed. For our future work, we would first focus on either incorporating an action recognition framework for this or exploring the modular ability of HSMMs by increasing the number of Gaussian Components to learn different interactions using a single HSMM, which can then be used for looking into trajectory segmentation as in [24], [25]. Moreover, we plan to incorporate Inverse Kinematics to better improve the physical contact during the interaction, rather than looking just at the joint space trajectories.

ACKNOWLEDGEMENTS

The authors would like to thank Mark Baierl, Michel Kohl and Martina Gassen whose work helped in developing of parts of the approach.

REFERENCES

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, 2018.

[2] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[3] P. Becker, H. Pandya, G. Gebhardt, C. Zhao, C. J. Taylor, and G. Neumann, "Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces," in *International Conference on Machine Learning (ICML)*, 2019.

[4] S. Bitzer and S. Vijayakumar, "Latent spaces for dynamic movement primitives," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2009.

[5] J. Bütepage, A. Ghadirzadeh, Ö. Ö. Karadag, M. Björkman, and D. Kragic, "Imitating by generating: Deep generative models for imitation of interactive tasks," *Frontiers in Robotics and AI*, 2020.

[6] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent service robotics*, 2016.

[7] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, "Learning collaborative manipulation tasks by demonstration using a haptic interface," in *International Conference on Advanced Robotics (ICAR)*, 2009.

[8] J. Campbell and H. B. Amor, "Bayesian interaction primitives: A slam approach to human-robot interaction," in *Conference on Robot Learning (CoRL)*, 2017.

[9] M. Chaveroche, A. Malaisé, F. Colas, F. Charpillet, and S. Ivaldi, "A variational time series feature extractor for action prediction," *arXiv preprint arXiv:1807.02350*, 2018.

[10] N. Chen, J. Bayer, S. Urban, and P. Van Der Smagt, "Efficient movement representation by embedding dynamic movement primitives in deep autoencoders," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2015.

[11] N. Chen, M. Karl, and P. Van Der Smagt, "Dynamic movement primitives in latent space of time-dependent variational autoencoders," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2016.

[12] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[13] O. Dermy, M. Chaveroche, F. Colas, F. Charpillet, and S. Ivaldi, "Prediction of human whole-body movements with ae-promps," in *IEEE-RAS 18th International Conference on Humanoid Robots (HUMANOIDS)*, 2018.

[14] G. Dorta, S. Vicente, L. Agapito, N. D. Campbell, and I. Simpson, "Structured uncertainty prediction networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: object-lifting case study with a humanoid," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2009.

[16] M. Ewerton, G. Neumann, R. Lioutikov, H. B. Amor, J. Peters, and G. Maeda, "Learning multiple collaborative tasks with a mixture of interaction primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[17] L. Fritsche, F. Unverzag, J. Peters, and R. Calandra, "First-person tele-operation of a humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2015.

[18] J. Han, M. R. Min, L. Han, L. E. Li, and X. Zhang, "Disentangled recurrent wasserstein autoencoder," in *International Conference on Learning Representations (ICLR)*, 2021.

[19] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear algebra and its applications*, 1988.

[20] M. Karl, M. Soelch, J. Bayer, and P. Van der Smagt, "Deep variational bayes filters: Unsupervised learning of state space models from raw data," in *International Conference on Learning Representations (ICLR)*, 2017.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[22] D. Koert, J. Pajarinen, A. Schotschneider, S. Trick, C. Rothkopf, and J. Peters, "Learning intention aware online adaptation of movement primitives," *IEEE Robotics and Automation Letters*, 2019.

[23] D. Koert, S. Trick, M. Ewerton, M. Lutter, and J. Peters, "Online learning of an open-ended skill library for collaborative tasks," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2018.

[24] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," *The International Journal of Robotics Research (IJRR)*, 2017.

[25] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters, "Probabilistic segmentation applied to an assembly task," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2015.

[26] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2014.

[27] G. Maeda, M. Ewerton, G. Neumann, R. Lioutikov, and J. Peters, "Phase estimation for fast action recognition and trajectory generation in human–robot collaboration," *The International Journal of Robotics Research (IJRR)*, 2017.

[28] G. Maeda, M. Ewerton, T. Osa, B. Busch, and J. Peters, "Active incremental learning of robot movement primitives," in *Conference on Robot Learning*, 2017.

[29] M. Nagano, T. Nakamura, T. Nagai, D. Mochihashi, I. Kobayashi, and W. Takano, "Hvgh: unsupervised segmentation for high-dimensional time series using deep neural compression and statistical generative model," *Frontiers in Robotics and AI*, 2019.

[30] A. K. Pandey and R. Gelin, "A mass-produced sociable humanoid robot: Pepper: The first machine of its kind," *IEEE Robotics & Automation Magazine*, 2018.

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

[32] E. Pignat and S. Calinon, "Learning adaptive dressing assistance from human demonstration," *Robotics and Autonomous Systems (RAS)*, 2017.

[33] V. Prasad, R. Stock-Homburg, and J. Peters, "Learning human-like hand reaching for human-robot handshaking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[34] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International Conference on Machine Learning (ICML)*, 2014.

[35] L. Rozo, J. Silverio, S. Calinon, and D. G. Caldwell, "Learning controllers for reactive and proactive behaviors in human–robot collaboration," *Frontiers in Robotics and AI*, 2016.

[36] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive motion of animals and machines*, 2006.