

# MoVEInt: Mixture of Variational Experts for Learning Human-Robot Interactions from Demonstrations

Vignesh Prasad<sup>1</sup>, Alap Kshirsagar<sup>1</sup>, Dorothea Koert<sup>1,3</sup>  
Ruth Stock-Homburg<sup>2</sup>, Jan Peters<sup>1,3,4,5</sup>, Georgia Chalvatzaki<sup>1,5</sup>

**Abstract**—Shared dynamics models are important for capturing the complexity and variability inherent in Human-Robot Interaction (HRI). Therefore, learning such shared dynamics models can enhance coordination and adaptability to enable successful reactive interactions with a human partner. In this work, we propose a novel approach for learning a shared latent space representation for HRIs from demonstrations in a Mixture of Experts fashion for reactively generating robot actions from human observations. We train a Variational Autoencoder (VAE) to learn robot motions regularized using an informative latent space prior that captures the multimodality of the human observations via a Mixture Density Network (MDN). We show how our formulation derives from a Gaussian Mixture Regression formulation that is typically used approaches for learning HRI from demonstrations such as using an HMM/GMM for learning a joint distribution over the actions of the human and the robot. We further incorporate an additional regularization to prevent “mode collapse”, a common phenomenon when using latent space mixture models with VAEs. We find that our approach of using an informative MDN prior from human observations for a VAE generates more accurate robot motions compared to previous HMM-based or recurrent approaches of learning shared latent representations, which we validate on various HRI datasets involving interactions such as handshakes, fistbumps, waving, and handovers. Further experiments in a real-world human-robot handover scenario show the efficacy of our approach for generating successful interactions with four different human interaction partners.

**Index Terms**—Physical Human-Robot Interaction, Imitation Learning, Learning from Demonstration

## I. INTRODUCTION

**E**NSURING a timely response while performing an interaction can enable a feeling of connectedness to ones partner [1]. Therefore, learning to generate response motions

Manuscript received: November 30, 2023; Revised March 5, 2024; Accepted April 3, 2024. This paper was recommended for publication by Editor Aleksandra Faust upon evaluation of the Associate Editor and Reviewers’ comments. Corresponding Author: Vignesh Prasad (vignesh.prasad@tu-darmstadt.de) Additional resources such as codes and models can be found at <https://bit.ly/MoVEInt>. This research work has received funding from the German Research Foundation (DFG) Emmy Noether Programme (CH 2676/1-1), the German Federal Ministry of Education and Research (BMBF) Projects “IKIDA” (Grant no.: 01IS20045) and “KompAKI” (Grant no.: 02L19C150), the Excellence Program, “The Adaptive Mind”, of the Hessian Ministry of Higher Education, Science, Research and Art, the EU’s Horizon Europe projects “MANiBOT” (Grant no.: 101120823) and “ARISE” (Grant no.: 101135959).

<sup>1</sup> Department of Computer Science, TU Darmstadt, Germany

<sup>2</sup> Chair for Marketing and Human Resource Management (MuP), Department of Law and Economics, TU Darmstadt, Germany

<sup>3</sup> Centre for Cognitive Science, TU Darmstadt, Germany

<sup>4</sup> German Research Center for AI (DFKI), Darmstadt, Germany

<sup>5</sup> Hessian Center for Artificial Intelligence (hessian.AI), Darmstadt, Germany  
Digital Object Identifier (DOI): see top of this page.

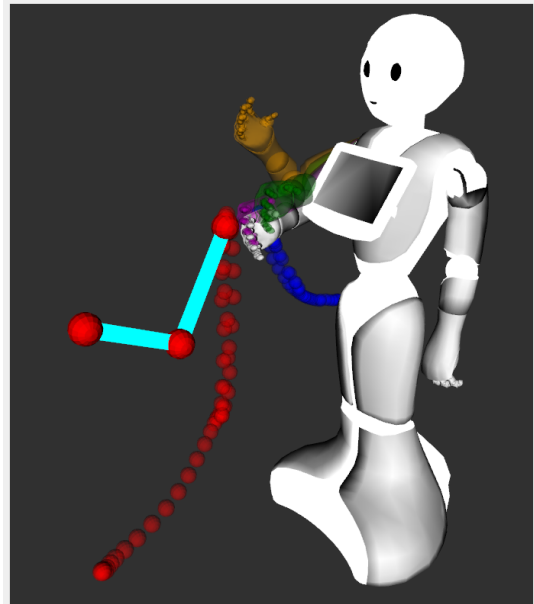


Fig. 1: Target poses generated in a reactive manner by the mixture of policies learned by MoVEInt for a Handshake interaction with the humanoid robot Pepper. MoVEInt generates multiple policies (shown in green, magenta, and orange) based on human observations which are combined to generate suitable robot motions.

for a robot in a timely manner for Human-Robot Interaction (HRI) is an important aspect of the interaction. One way to do so is by learning a shared representation space between the human and the robot [2]–[8]. An important aspect of such approaches, for learning HRI from demonstrations, is accurately capturing the multimodality of the underlying data to effectively capture the underlying skills and accurately generate response motions.

Vogt et al. [7] showed the use of shared latent spaces in an Imitation Learning (IL) approach by decomposing interactions into multiple segments in a lower dimensional space with an underlying Gaussian Mixture Model (GMM) and learning the sequence of key poses using a Hidden Markov Model (HMM) to define the progression of an interaction. In our previous work, MILD [5], [6], we explore learning a shared latent space model using a Variational Autoencoder (VAE) wherein we learn a joint distribution over the trajectories of both the human and the robot using an HMM with underlying Gaussian States to represent the multimodality of the demonstrations. Rather than extracting key poses as in [7], we generate the robot’s motion using Gaussian Mixture Regression (GMR) from the underlying HMM based on the human’s observations

in a reactive manner. In doing so, we achieve better accuracy than using a recurrent representation of the shared latent dynamics [2].

Generative approaches have been used to jointly learn human and robot policies for collaborative tasks [8]–[10] and discover different underlying latent “strategies” of the human but only for a given task. In our work, we further explore how underlying latent strategies can be learned from different tasks in a dataset rather than just a single task by using a mixture distribution to predict different latent policies, which are then combined in a Mixture of Experts fashion. An example of this can be seen in Fig. 1, where we show a handshake interaction with the Pepper robot. Trained on a dataset of different physical interactions like waving, handshakes and fistbumps, we see the different policies that get predicted (the reconstructions of which are shown by the different colored arms of Pepper) which are then combined in the latent space yielding a suitable response motion.

To learn multiple latent policies and effectively combine them, we employ Mixture Density Networks (MDNs) [11] to capture the multimodality of the demonstrations. MDNs predict a mixture of Gaussians and the corresponding mixture coefficients yielding a multimodal prediction, rather than a unimodal distribution or a single output. MDN policy representations have been widely used in Imitation Learning and Reinforcement Learning (RL) on a variety of tasks such as autonomous driving [12], [13], entropy regularization for imitation learning [14], adapting to multiple goals [15], incorporating human intentions for robot tasks [16], predicting gaze behaviors for robot manipulation [17] or more generally as improved attention mechanisms [18], [19] or as forward models of the environment [20]–[23].

With these key ideas for learning shared multimodal latent policy representations for HRI, the main contributions of our paper are as follows. We propose “MoVEInt”, a novel framework that employs a Mixture of Variational Experts for learning Human-Robot Interactions from demonstrations through a shared latent representation of a human and a robot. We learn latent space policies in a Mixture of Experts fashion via a Mixture Density Network (MDN) to encode the latent trajectory of a human partner, regularize the robot embeddings, and subsequently, predict the robot motions reactively. We show how our proposed formulation extends from Gaussian Mixture Regression (GMR) which is typically used in HMM-based approaches for learning HRI from demonstrations. We leverage the function approximation powers of Neural Networks to simplify the GMR formulation to a linear Mixture of Gaussian formulation.

Through our experiments, we see that our approach successfully captures the best of recurrent, multi-modal, and reactive representations for learning short-horizon Human-Robot Interactions from demonstrations. We find that MoVEInt generates highly accurate robot behaviors without explicit action labels, which is more natural as humans also internally infer what our interaction partner is doing and adapt to it without explicitly communicating the action being done. We validate our predictive performance on a variety of physical HRI scenarios such as handshakes, fistbumps, and robot-to-human handovers. We

further demonstrate the efficacy of MoVEInt on a real-world interaction scenario for bimanual (dual-arm) robot-to-human handovers.

## II. FOUNDATIONS

### A. Variational Autoencoders

Variational Autoencoders (VAEs) [24], [25] are a class of neural networks that learn to reconstruct inputs via latent representations in an unsupervised, probabilistic way. The inputs “ $\mathbf{x}$ ” are encoded into lower dimensional latent embeddings “ $\mathbf{z}$ ” that a decoder uses to reconstruct the input. A prior distribution, typically a standard normal distribution  $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ , is enforced over the latent space during training. The goal is to estimate the true posterior  $p(\mathbf{z}|\mathbf{x})$  using a neural network  $q(\mathbf{z}|\mathbf{x})$ , which is trained by minimizing the Kullback-Leibler (KL) divergence between them

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

which can be re-written as

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

The KL divergence is always non-negative; therefore, the second term in Eq. 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

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

The first term aims to reconstruct the input via samples decoded from the posterior. The second term is the KL divergence between the prior and the posterior, which regularizes the learning. Further information can be found in [24]–[26].

### B. Mixture Density Networks

Mixture Density Networks (MDNs) [11] is a probabilistic neural network architecture that encapsulates the representation powers of neural networks with the modular advantages that come with Gaussian Mixture Models (GMMs). MDNs parameterize a typical supervised regression problem of predicting an output distribution  $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}^2(\mathbf{x}))$  as a multimodal distribution by predicting a set of means and variances  $(\boldsymbol{\mu}_i(\mathbf{x}), \boldsymbol{\sigma}_i^2(\mathbf{x}))$  corresponding to the mixture distribution along with the mixture coefficients  $\alpha_i(\mathbf{x})$  corresponding to the relative weights of the mixture distributions.

## III. MIXTURE OF VARIATIONAL EXPERTS FOR LEARNING HUMAN-ROBOT INTERACTIONS FROM DEMONSTRATIONS

In this section, we present MoVEInt, a novel framework that learns latent space policies in a Mixture of Experts fashion for modeling the shared dynamics of a human and a robot in HRI tasks. This process can be seen in Fig. 2. We aim to model the dynamics of HRI tasks via shared latent representations of a human and a robot in a way that captures the multimodality

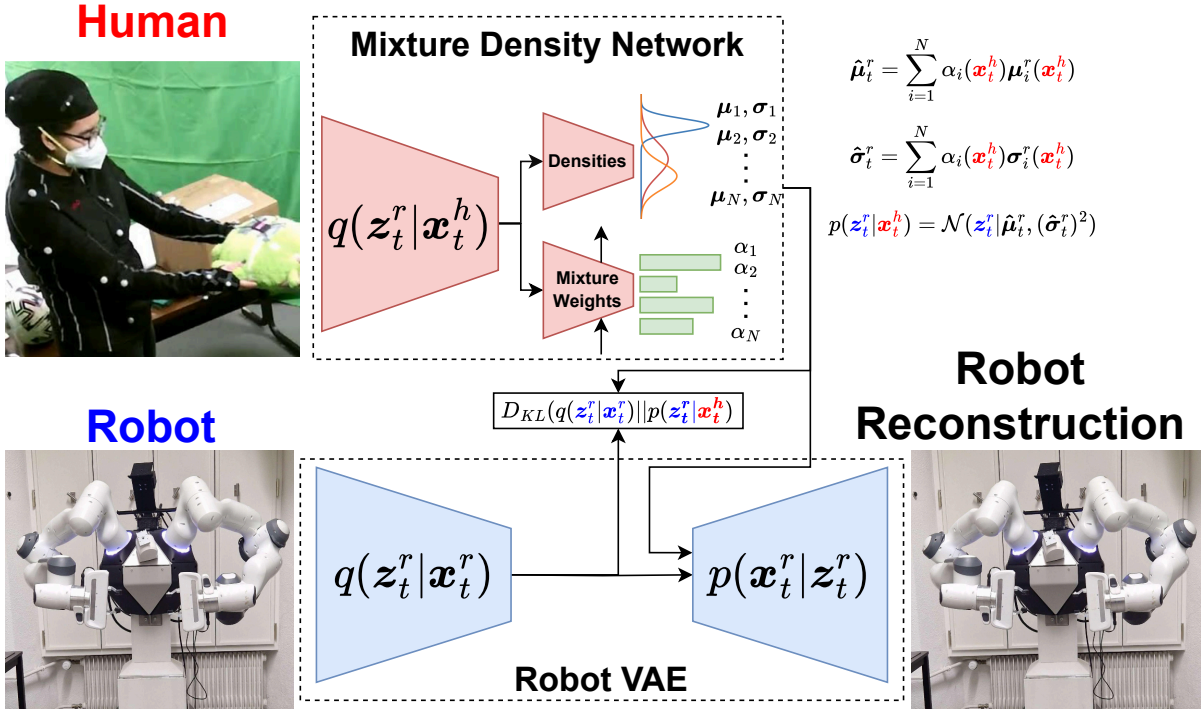


Fig. 2: Overview of our approach “MoVEInt”. We train a reactive policy using a Mixture Density Network (MDN) to predict latent space robot actions from human observations. The MDN policy is used not just for reactively generating the robot’s actions, but also to regularize a VAE that learns a latent representation of the robot’s actions. This regularization ensures that the learned robot representation matches the predicted MDN policy and also ensures that the robot VAE learns to decode samples from the MDN policy.

of the demonstrations and subsequently predicts the robot’s motions in a reactive manner. To do so, we use an MDN that takes the human observations as input and predicts multiple latent policies, thereby enabling a multimodal output, and subsequently, the relative weights for each policy so that they can be effectively combined. For learning a shared latent representation between the human and the robot, we train a VAE over the robot motions and regularize the VAE with the predicted policy from the MDN, thereby learning the robot embeddings and the subsequent human-conditioned policy predictions in a cohesive manner.

We motivate the use of MDNs in Sec. III-A by showing the equivalence of MDNs with Gaussian Mixture Regression (GMR) for HRI. We then explain learning the robot motion embeddings (Sec. III-B), and then show how to train reactive policies for HRI (Sec. III-C). We denote the human variables in **red** with the superscript  $h$  and the robot variables in **blue** with the superscript  $r$ .

#### A. GMR-based Interaction Dynamics with MDNs

Learning a joint distribution over the degrees of freedom of a human and a robot has been widely used in learning HRI from demonstrations [5], [6], [27]–[31]. With a joint distribution, GMR provides a mathematically sound formulation of predicting the conditional distribution of the robot actions. When using a Mixture of  $N$  Gaussian components  $\{\mu_i, \Sigma_i\}$  that model a joint distribution of the Human and Robot trajectories, the distribution can be decomposed into the marginals for the human and the robot

$$\mu_i = \begin{bmatrix} \mu_i^h \\ \mu_i^r \end{bmatrix}; \Sigma_i = \begin{bmatrix} \Sigma_i^{hh} & \Sigma_i^{hr} \\ \Sigma_i^{rh} & \Sigma_i^{rr} \end{bmatrix} \quad (4)$$

In such a formulation, during test time, the robot motions can be generated reactively by conditioning the distribution using Gaussian Mixture Regression (GMR) [27], [32]<sup>1</sup>

$$\begin{aligned} K_i &= \Sigma_i^{rr} (\Sigma_i^{hh})^{-1} \\ \hat{\mu}_i^r &= \mu_i^r + K_i (\mathbf{z}_t^h - \mu_i^h) \\ \hat{\mu}_t^r &= \sum_{i=1}^N \alpha_i(\mathbf{z}_t^h) \hat{\mu}_i^r \\ \hat{\Sigma}_i^r &= \Sigma_i^{rr} - K_i \Sigma_i^{hr} + (\hat{\mu}_i^r - \mu_i^r) (\hat{\mu}_i^r - \mu_i^r)^T \\ \hat{\Sigma}_t^r &= \sum_{i=1}^N \alpha_i(\mathbf{z}_t^h) \hat{\Sigma}_i^r \\ p(\mathbf{z}_t^r | \mathbf{z}_t^h) &= \mathcal{N}(\mathbf{z}_t^r | \hat{\mu}_t^r, \hat{\Sigma}_t^r) \end{aligned} \quad (5)$$

where  $\alpha_i(\mathbf{z}_t^h)$  is the relative weight of each component.

Revisiting Eq. 5, given the linear dependence of  $\hat{\mu}_i^r$  on  $\mathbf{z}_t^h$ , we can therefore consider  $\hat{\mu}_i^r$  as a direct output of a neural network. Although the covariance matrix has a quadratic relationship with  $\mathbf{z}_t^h$ , considering Neural Networks are powerful function approximations, we assume that our network can adequately approximate a diagonalized form of the covariance matrices  $\hat{\Sigma}_i^r$  [15], [33]. To consider the temporal aspect of learning such trajectories from demonstrations, rather than using the Mixture Model coefficients,  $\alpha_i(\mathbf{z}_t^h)$  can be approximated in a temporal manner using Hidden Markov Models (HMMs) [27]  $\alpha_i(\mathbf{z}_t^h) = \mathcal{N}(\mathbf{z}_t^h; \mu_i, \Sigma_i) \sum_{j=1}^N \alpha_j(\mathbf{z}_{t-1}) T_{j,i}$  where the parameters  $(\mu_i, \Sigma_i, T_{j,i})$  are the means and co-

<sup>1</sup>The covariance in the GMR Formulation in [27], which is derived from [32], can be obtained by simplifying Eq. 5 as shown in the appendix.

variances of the underlying Gaussian states and the transition probabilities between the states respectively.

Given that there exist parallels between HMMs and RNNs [34]–[36] and to capture the recurrence in predicting  $\alpha_i(\mathbf{z}_t^h)$ , we use a recurrent layer for predicting the mixing coefficients. Using the predictions of the mixture model parameters  $(\boldsymbol{\mu}_i^r(\mathbf{x}_t^h), \boldsymbol{\sigma}_i^r(\mathbf{x}_t^h)^2)$  and coefficients  $\alpha_i(\mathbf{x}_t^h)$  from the MDN<sup>2</sup>, we can re-write Eq. 5 as

$$\begin{aligned}\hat{\boldsymbol{\mu}}_t^r &= \sum_{i=1}^N \alpha_i(\mathbf{x}_t^h) \boldsymbol{\mu}_i^r(\mathbf{x}_t^h) \\ (\hat{\boldsymbol{\sigma}}_t^r)^2 &= \sum_{i=1}^N \alpha_i(\mathbf{x}_t^h) \boldsymbol{\sigma}_i^r(\mathbf{x}_t^h)^2 \\ p(\mathbf{z}_t^r | \mathbf{x}_t^h) &= \mathcal{N}(\mathbf{z}_t^r | \hat{\boldsymbol{\mu}}_t^r, (\hat{\boldsymbol{\sigma}}_t^r)^2)\end{aligned}\quad (6)$$

which is then used as a prior for regularizing the VAE and training the decoder to reconstruct latent samples obtained after observing the human partner.

### B. Robot Motion Embeddings

To learn a meaningful representation of the robot’s actions, we train a VAE to reconstruct the robot’s actions at each timestep. Typically, in VAEs, a standard normal distribution is used as the latent space prior  $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Rather than forcing an uninformative standard normal prior as in Eq. 3, we use the reactive policy predicted from the human observations by the MDN (Eq. 6) to regularize the VAE’s posterior  $KL(q(\mathbf{z}_t^r | \mathbf{x}_t^r) || p(\mathbf{z}_t^r | \mathbf{x}_t^h))$ , thereby learning a task-oriented latent space that is in line with the interaction dynamics. Our ELBO for training the robot VAE can be written as

$$\begin{aligned}ELBO_t^r &= \mathbb{E}_{q(\mathbf{z}_t^r | \mathbf{x}_t^r)} [\log p(\mathbf{x}_t^r | \mathbf{z}_t^r)] \\ &\quad - \beta KL(q(\mathbf{z}_t^r | \mathbf{x}_t^r) || p(\mathbf{z}_t^r | \mathbf{x}_t^h))\end{aligned}\quad (7)$$

Where  $\beta$  is a relative weight used to ensure numerical stability between the KL divergence term and the image reconstruction term [26].

### C. Reactive Motion Generation

We aim to learn a policy for reactively generating the robot’s latent trajectory based on human observations  $p(\mathbf{z}_t^r | \mathbf{x}_t^h)$ . We do so in a Behavior Cloning Paradigm by maximizing the probability of the observed trajectories w.r.t. the current policy  $\mathcal{L}_t^{BC} = -\mathbb{E}_{\mathbf{z}_t^r \sim p(\mathbf{z}_t^r | \mathbf{x}_t^h)} p(\mathbf{x}_t^r | \mathbf{z}_t^r)$  wherein we first draw samples from the current policy  $\mathbf{z}_t^r \sim p(\mathbf{z}_t^r | \mathbf{x}_t^h)$  which we then reconstruct  $p(\mathbf{x}_t^r | \mathbf{z}_t^r)$ , thereby enabling the decoder to reconstruct latent samples obtained after observing the human, as done during test time.

However, as highlighted in [15], MDN policy representations are prone to mode collapses. Therefore, to ensure adequate separation between the modes so that we can learn a diverse range of actions, we employ a contrastive loss at each timestep. The contrastive loss pushes the means of each

mixture component further away, while maintaining temporal similarity by pushing embeddings that are closer in time nearer to each other. Further, as done in [15], we add entropy cost to ensure a balanced prediction of the mixture coefficients. Our separation loss can be written as

$$\begin{aligned}\mathcal{L}_t^{sep} &= \underbrace{\sum_{i=1}^{N-1} \sum_{j=i+1}^N \exp(-\|\boldsymbol{\mu}_i^r(\mathbf{x}_t^h) - \boldsymbol{\mu}_j^r(\mathbf{x}_t^h)\|^2)}_{\text{separation of means}} \\ &\quad + 1 - \underbrace{\frac{1}{N} \sum_{i=1}^N \exp(-\|\boldsymbol{\mu}_i^r(\mathbf{x}_t^h) - \boldsymbol{\mu}_i^r(\mathbf{x}_{t-1}^h)\|^2)}_{\text{temporal closeness}} \\ &\quad + \underbrace{\sum_{i=1}^N \alpha_i(\mathbf{x}_t^h) \ln \alpha_i(\mathbf{x}_t^h)}_{\text{entropy cost}}\end{aligned}\quad (8)$$

Our final loss consists of the Behavior Cloning loss, the ELBO of the robot VAE, and the separation loss

$$\sum_{t=1}^T [\mathcal{L}_t^{BC} - ELBO_t^r + \beta \mathcal{L}_t^{sep}] \quad (9)$$

where  $\beta$  is the same KL weight factor used in Eq. 7. Our overall training procedure is shown in Alg. 1.

---

#### Algorithm 1: Learning a Reactive Latent Policy for Human-Robot Interaction

---

**Data:** A set of human and robot trajectories

$$\mathbf{X} = \{\mathbf{X}^h, \mathbf{X}^r\}$$

**Result:** MDN Policy Network, Robot VAE

**while not converged do**

**for**  $\mathbf{x}_{1:T}^h, \mathbf{x}_{1:T}^r \in \mathbf{X}$  **do**

        Compute the MDN policy  $p(\mathbf{z}_t^r | \mathbf{x}_t^h)$  (Eq. 6)

        Compute the robot VAE posterior  $p(\mathbf{z}_t^r | \mathbf{x}_t^r)$  (Eq. 7)

        Reconstruct samples from  $p(\mathbf{z}_t^r | \mathbf{x}_t^h)$  and  $p(\mathbf{z}_t^r | \mathbf{x}_t^r)$

        Minimize the loss in Eq. 9 to update the network

**end**

**end**

---

During test time, given human observations  $\mathbf{x}_t^h$ , we compute the latent policy from the MDN  $\mathbf{z}_t^r = p(\mathbf{z}_t^r | \mathbf{x}_t^h)$  which is then decoded to obtain the robot action  $p(\mathbf{x}_t^r | \mathbf{z}_t^r)$ .

## IV. EXPERIMENTS AND RESULTS

In this section, we first explain the datasets we use (Sec. IV-A), then we highlight the implementation details (Sec. IV-B), and finally present our results (Sec. IV-C).

### A. Dataset

1) *Bütepage et al.*: [2] record HHI and HRI demonstrations of 4 interactions: Waving, Handshaking, and two kinds of fist bumps. The first fistbump, called “Rocket Fistbump”, involves bumping fists at a low level and then raising them upwards while maintaining contact with each other. The second is called “Parachute Fistbump” in which partners bump their fists at a high level and bring them down while simultaneously

<sup>2</sup>Since the relation between  $\mathbf{x}_t^h$  and  $\mathbf{z}_t^h$  is deterministic, for ease of notation, we show the prediction of the MDN components with  $\mathbf{x}_t^h$

oscillating the hands sideways, while in contact with each other. They additionally record demonstrations of these actions with a human partner interacting with an ABB YuMi-IRB 14000 robot controlled via kinesthetic teaching. We call this scenario as “HRI-Yumi”.

We further adapt the HHI demonstrations for the humanoid robot Pepper [37] by extracting the joint angles from one of the human partner’s skeletons from the HHI data for the Pepper robot using the similarities in DoFs between a human and Pepper [38], [39], which we denote this as “HRI-Pepper”. Given the smaller number of trajectories in the HRI-Yumi scenario, we initialize the network with the weights from the model trained for the HHI scenario.

2) *Nuitrack Skeleton Interaction Dataset*: (NuiSI) [6] is a dataset that we collected ourselves of the same 4 interactions as in [2]. The skeleton data of two human partners interacting with one another is recorded using two Intel Realsense D435 cameras (one recording each human partner). We use Nuitrack [40] for tracking the upper body skeleton joints in each frame at 30Hz. As done with [2], we show results on both the HHI data as well as on the joint angles extracted for the Pepper robot from the skeleton trajectories for an HRI-Pepper scenario.

3) *Human-Human Object Handovers Dataset*: [41] is a dataset consisting of 12 pairs of participants performing object handovers with various objects, with each partner taking the role of the giver and the receiver, leading to 24 pairs of handover partners. The dataset consists of both Unimanual (single-arm) and Bimanual (dual-arm) handovers tracked using motion capture at 120Hz, which we downsample to 30Hz. We explore a Robot-to-Human handover scenario and use the giver’s observations corresponding to the robot and the receiver’s observations corresponding to the human.

Dataset	Downsampled FPS (Hz)	No. of Trajectories	
		Training	Testing
HHI, HRI-Pepper [2]	20	149	32
HRI-Yumi [2]	20	32	9
HHI, HRI-Pepper (NuiSI) [6]	30	33	11
HHI-Handovers [41]	30	168	24

TABLE I: Statistics of the different datasets used.

### B. Implementation Details

After downsampling the data to the frequencies mentioned in Table I, we use a time window of observations as the input for a given timestep, similar to [2]. We use the 3D positions and the velocities (represented as position deltas) of the right arm joints (shoulder, elbow, and wrist), with the origin at the shoulder, leading to an input size of 90 dimensions (5x3x6: 5 timesteps, 3 joints, 6 dimensions) for a human partner. For the HHI-Handover scenario, we use both the left and right arm data (180 dimensions). For the HRI-Pepper and HRI-Yumi scenarios, we use a similar window of joint angles, leading to an input size of 20 dimensions (5x4) for the 4 joint angles of Pepper’s right arm and an input size of 35 dimensions (5x7) for the 7 joint angles of Yumi’s right arm. The reconstruction loss for the VAE and Behavior Cloning are calculated by decoding samples drawn from the respective distributions in a Monte Carlo fashion.

For the dataset from [2] and the NuiSI dataset [6], we use a VAE with 2 hidden layers each in the encoder and decoder with a dimensionality of (40, 20) and (20, 40) respectively, with Leaky ReLU activations and a 5D latent space. For the VAE posterior and the MDN outputs, we predict the mean and the log of the standard deviation (logstd). We add a regularization of  $10^{-6}$  to the standard deviation. The MDN has a similar structure as the VAE encoder but predicts 3 different means and logstds. For the mixture coefficients  $\alpha_i$ , to enable a recurrent nature of the predictions, the output of the MDN encoder is passed to a single-layer Gated Recurrent Unit (GRU) whose outputs are then passed through a linear layer followed by a softmax layer. For the NuiSI dataset, we initialize the model with the pre-trained weights trained on the dataset in [2]. For the Handover dataset [41], we use double the number of states in each layer, namely 80 and 40 hidden states and a 10D latent space.

### C. Reactive Motion Generation Results

To show the advantage of our approach, we compare MoVEInt with Bütepage *et al.* [2], who use an LSTM-based approach as a latent regularization for reactive motion generation, which is close to a unimodal version of our approach. Further, we compare MoVEInt to MILD [6] which uses HMMs to capture the multimodal latent dynamics of interactive tasks. The efficacy of MoVEInt can be seen via the low mean squared error of the predicted robot motions (Table II).

We perform better than both MILD [6] and Bütepage *et al.* [2] on almost all interaction scenarios. We additionally want to highlight that on the HRI-Pepper scenarios, unlike MILD [6] and Bütepage *et al.* [2] where the pre-trained model from the HHI scenario is used, we train our model completely from scratch and still achieve better performance. Moreover, it is worth noting that both [2] and MILD are trained in a partially supervised manner using the interaction labels. In [2], a one-hot label denoting the interaction being performed is given as an input to the network for generalizing to different interactions, whereas in MILD, a separate HMM is trained for each interaction. In contrast, MoVEInt is trained on all the tasks in a given dataset without any labels in a purely unsupervised manner and still achieves competitive results on the different datasets.

We additionally show some quantitative results of MoVEInt. We train a Handover model with just the hand trajectories whose predictions are used for reactive motion generation on a Bimanual Franka Emika Panda robot setup, “Kobo”, as shown in Fig. 3. Some additional examples of the trajectories generated by MoVEInt for bimanual and unimanual handovers from the HHI-Handovers dataset [41] are shown in Fig. 4a and 4b respectively. Since MoVEInt is trained on all the interactions in a corresponding dataset, which, when coupled with the separation loss, learns a diverse and widespread set of components that cover the various demonstrations, as can be seen by the reconstruction of the individual components. Combining the components in the latent space subsequently leads to an accurate and suitable motion generated reactively



Dataset (units)	Action	MILD [6]	Bütepage et al. [2]	MoVEInt
HHI (Bütepage et al. [2]) (cm)	Hand Wave	0.788 ± 1.226	4.121 ± 2.252	<b>0.448 ± 0.630</b>
	Handshake	1.654 ± 1.549	1.181 ± 0.859	<b>0.196 ± 0.153</b>
	Rocket Fistbump	0.370 ± 0.682	0.544 ± 1.249	<b>0.123 ± 0.175</b>
	Parachute Fistbump	0.537 ± 0.579	0.977 ± 1.141	<b>0.314 ± 0.348</b>
HRI-Pepper (Bütepage et al. [2]) (rad)	Hand Wave	0.103 ± 0.103	0.664 ± 0.277	<b>0.087 ± 0.089</b>
	Handshake	0.056 ± 0.041	0.184 ± 0.141	<b>0.015 ± 0.014</b>
	Rocket Fistbump	0.018 ± 0.035	0.033 ± 0.045	<b>0.007 ± 0.015</b>
	Parachute Fistbump	0.088 ± 0.148	0.189 ± 0.196	<b>0.048 ± 0.112</b>
HRI-Yumi (Bütepage et al. [2]) (rad)	Hand Wave	1.033 ± 1.204	0.225 ± 0.302	<b>0.147 ± 0.072</b>
	Handshake	0.068 ± 0.052	0.133 ± 0.214	<b>0.057 ± 0.044</b>
	Rocket Fistbump	0.128 ± 0.071	0.147 ± 0.119	<b>0.093 ± 0.045</b>
	Parachute Fistbump	<b>0.028 ± 0.034</b>	0.181 ± 0.155	0.081 ± 0.082
HHI (NuiSI [6]) (cm)	Hand Wave	0.408 ± 0.538	3.168 ± 3.392	<b>0.298 ± 0.274</b>
	Handshake	0.311 ± 0.259	1.489 ± 3.327	<b>0.149 ± 0.120</b>
	Rocket Fistbump	1.142 ± 1.375	3.576 ± 3.082	<b>0.673 ± 0.679</b>
	Parachute Fistbump	0.453 ± 0.578	2.008 ± 2.024	<b>0.291 ± 0.199</b>
HRI-Pepper (NuiSI [6]) (rad)	Hand Wave	0.046 ± 0.059	0.057 ± 0.093	<b>0.044 ± 0.048</b>
	Handshake	0.020 ± 0.014	0.083 ± 0.075	<b>0.011 ± 0.008</b>
	Rocket Fistbump	0.077 ± 0.067	0.101 ± 0.086	<b>0.045 ± 0.045</b>
	Parachute Fistbump	0.022 ± 0.027	0.049 ± 0.040	<b>0.017 ± 0.014</b>
HHI-Handovers (Kshirsagar et al. [41]) (cm)	Unimanual	0.441 ± 0.280	1.133 ± 0.721	<b>0.441 ± 0.221</b>
	Bimanual	0.869 ± 0.964	0.990 ± 0.764	<b>0.685 ± 0.643</b>

TABLE II: Prediction MSE for robot trajectories after observing the human partner averaged over all joints and timesteps. Results for the HHI scenarios are in cm and for the HRI scenarios are in radians. (Lower is better)

(shown in blue). For further examples, please refer to our supplementary video.

#### D. User study

To understand the effectiveness of our approach in the real world, we perform a feasibility study as a proof-of-concept with five users who perform bimanual robot-to-human handovers with the Kobo robot. We evaluate the ability of our approach to successfully generate a handover motion with three different objects where each participant interacts with the robot five times for each object (a total of 15 runs per participant). To maintain the object-centric nature of the interaction, we use a controller that tracks the mid-point of both end-effectors, thereby resembling tracking the object’s trajectory. This study was approved by the Ethics Commission at TU Darmstadt (Application EK 20/2023). For the study, we modify the Handovers dataset [41] such that the giver’s trajectories fall within the task space limits of the Kobo robot. We train a separate model to use only the receiver’s hand trajectories and subsequently predict the end-effector trajectory for the robot. This is given to an Object-centric Inverse Kinematics controller that tracks the motion of the mid-point between the end-effectors.

As shown in Table III, our approach can generate successful handover trajectories for different users and different objects. We observed some failure cases due to sudden jumps in the predicted robot motions resulting from inaccuracies in perceiving the human, which would overshoot the robot’s dynamic limits. However, this failure could be avoided by incorporating additional filters over the input and output data to MoVEInt. Some failures occurred because the object did not reach the exact vicinity of the human’s hand location. This failure could be avoided by incorporating object-related information such as the size or weight, allowing the robot to gauge better when the handover is executed. Sometimes, the

robot would retreat before the human could grasp the object if sufficient time had passed. One reason for this hasty retreating behavior could be that the recurrent network’s hidden inputs overpower the observational input, causing the robot to follow the general motion of the handover seen during training and retreat. Such a failure could be mitigated by incorporating the robot state as part of the input.

Object \ User ID	User ID				Total (per object)
	#1	#2	#3	#4	
Stool	5	4	5	5	19/20
Box	4	5	4	4	17/20
Bedsheet	4	5	3	3	15/20
Total (per user)	13/15	14/15	12/15	12/15	<b>51/60</b>

TABLE III: Number of successful handovers of each object by the Kobo robot to each user (total of 5 per object per user i.e. total of 20 per object and 15 per user).

## V. CONCLUSION AND FUTURE WORK

In this work, we presented “MoVEInt”, a novel deep generative Imitation Learning approach for learning Human-Robot Interaction from demonstrations in a Mixture of Experts fashion. We demonstrated the use of Mixture Density Networks (MDNs) as a multimodal policy representation in a shared latent space of the human and the robot. We showed how MoVEInt stems from the GMR-based formulation of predicting interaction dynamics used in HMM-based approaches to learning HRI. We showed how our MDN policy can predict multiple underlying policies and combine them to effectively generate response motions for the robot. We verified the efficacy of MoVEInt across a variety of interactive tasks, where we found that MoVEInt mostly outperformed other baselines that either use explicitly modular representations like an HMM or simple recurrent policy representations. Our experimental evaluation showcases the versatility of MoVEInt, which effectively combines explicitly modular distributions

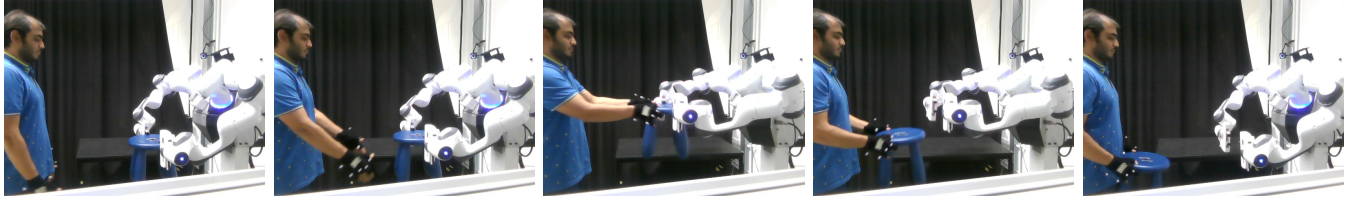


Fig. 3: Sample Human-Robot Interactions generated with the reactive motions generated by MoVEInt for a Bimanual Handover scenario.

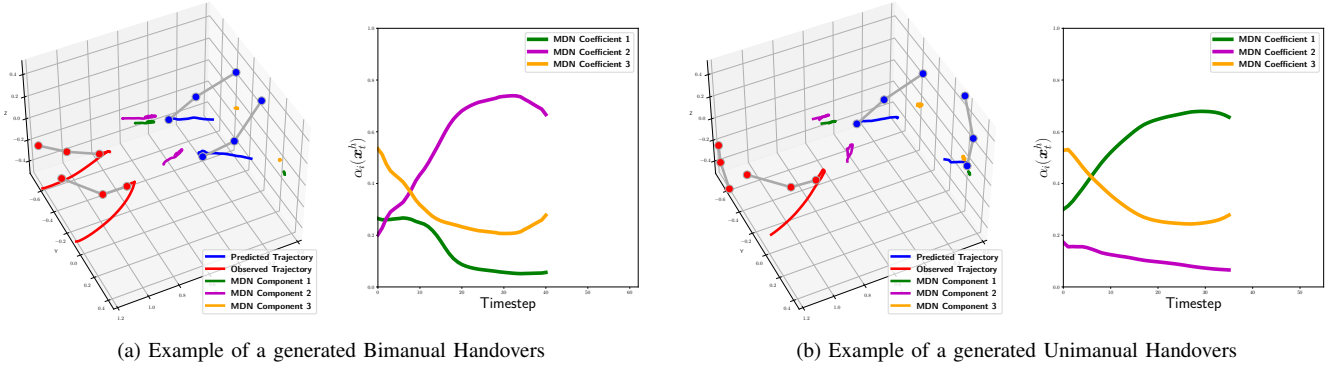


Fig. 4: Sample trajectories generated by MoVEInt for the Bimanual and Unimanual Handovers in the HHI-Handovers dataset in [41]. The 3D plots show the reconstructed trajectories and the 2D plots show the corresponding progression of  $\alpha_i(x_i^h)$  for the different components of the MDN. In the 3D plots, the observed trajectory of the receiver is shown in red and the generated trajectory of the giver is shown in blue and the giver’s corresponding ground truth is shown in black. The reconstruction of the individual latent components of the MDN are shown in green, magenta, and orange. It can be seen that the learned components correspond to different parts of the task space. For example, green denotes the hand locations for a unimanual handover, magenta denotes the hand locations for a bimanual handover, and orange denotes the static hand locations for the starting and ending neutral poses. In the 2D plot, it can be seen how the coefficients for components corresponding to bimanual (magenta) and unimanual (green) get activated based on the interaction being performed, while the component corresponding to a neutral pose (orange) gets activated at the beginning of the interaction while both partners are static.

with recurrent policy representations for learning interaction dynamics.

The main focus of this article is to explore the use of MDNs as a latent policy representation for simplistic short-horizon interactions like handshakes, handovers, etc., and show the feasibility of our approach in learning various interactive behaviors. One drawback is that we currently do not explicitly study how robust the approach is to unknown behaviors of the human, or how it performs on more complicated tasks.

Our next steps would be exploring how our approach scales to longer horizon tasks such as bi-directional handovers, collaborative sequential manipulation, or proactive tasks where the robot leads the interaction. This would be validated via a broader user study which would show the efficacy of our approach. Some extensions for extrapolating to such tasks can be explored by using better Generative models [8], [10], [42] or adding explicit constraints for combining reactive motion generation and planning [43]. Our future work would also explore incorporating task-related constraints such as object information for handover grasps [44], [45], force information for enabling natural interactive behaviors [46]–[48], and ergonomic and safety constraints for a more user-friendly interaction [49], [50].

## REFERENCES

- [1] K. L. Marsh, M. J. Richardson, and R. C. Schmidt, “Social connection through joint action and interpersonal coordination,” *Topics in cognitive science*, vol. 1, no. 2, pp. 320–339, 2009.
- [2] J. Bütepage, A. Ghadirzadeh, Ö. Öztimur Karadağ, M. Björkman, and D. Kragic, “Imitating by generating: Deep generative models for imitation of interactive tasks,” *Frontiers in Robotics and AI*, 2020.
- [3] A. Jonnavittula and D. P. Losey, “Learning to share autonomy across repeated interaction,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [4] R. P. Joshi, J. P. Tarapore, and T. Shibata, “Electric wheelchair-humanoid robot collaboration for clothing assistance of the elderly,” in *IEEE International Conference on Human System Interaction (HSI)*, 2020.
- [5] V. Prasad, D. Koert, R. Stock-Homburg, J. Peters, and G. Chaltatzaki, “Mild: Multimodal interactive latent dynamics for learning human-robot interaction,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2022.
- [6] V. Prasad, L. Heitlinger, D. Koert, R. Stock-Homburg, J. Peters, and G. Chaltatzaki, “Learning multimodal latent dynamics for human-robot interaction,” *arXiv preprint arXiv:2311.16380*, 2023.
- [7] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. B. Amor, “A system for learning continuous human-robot interactions from human-human demonstrations,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [8] C. Wang, C. Pérez-D’Arpino, D. Xu, L. Fei-Fei, K. Liu, and S. Savarese, “Co-gail: Learning diverse strategies for human-robot collaboration,” in *Conference on Robot Learning (CoRL)*, 2022.
- [9] A. Tormos Llorente, V. Giménez Ábalos, M. Domènech Vila, D. Gnatyshak, S. Álvarez Napagao, and J. Vázquez Salceda, “Explainable agents adapt to human behaviour,” in *International Workshop on Citizen-Centric Multi-Agent Systems (CMAS)*, 2023.
- [10] P. Sengadu Suresh, Y. Gui, and P. Doshi, “Dec-airl: Decentralized adversarial irl for human-robot teaming,” in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.
- [11] C. M. Bishop, “Mixture density networks,” 1994.
- [12] A. Baheri, “Safe reinforcement learning with mixture density network, with application to autonomous driving,” *Results in Control and Optimization*, 2022.
- [13] S. Kuutti, S. Fallah, and R. Bowden, “Adversarial mixture density networks: Learning to drive safely from collision data,” in *IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

- [14] K. Lee, S. Choi, and S. Oh, "Maximum causal tsallis entropy imitation learning," *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [15] Y. Zhou, J. Gao, and T. Asfour, "Movement primitive learning and generalization: Using mixture density networks," *IEEE Robotics & Automation Magazine*, 2020.
- [16] M. Thabet, M. Patacchiola, and A. Cangelosi, "Sample-efficient deep reinforcement learning with imaginary rollouts for human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [17] H. Kim, Y. Ohmura, and Y. Kuniyoshi, "Memory-based gaze prediction in deep imitation learning for robot manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.
- [18] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," in *International Conference on Learning Representations (ICLR)*, 2016.
- [19] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Boloni, "Pay attention!-robustifying a deep visuomotor policy through task-focused visual attention," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [20] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [21] O. C. Schrempf and U. D. Hanebeck, "A generic model for estimating user intentions in human-robot cooperation," in *International Conference on Informatics in Control, Automation and Robotics. SCITEPRESS*, 2005.
- [22] H. Zhang, E. Heiden, S. Nikolaidis, J. J. Lim, and G. S. Sukhatme, "Auto-conditioned recurrent mixture density networks for learning generalizable robot skills," *arXiv preprint arXiv:1810.00146*, 2018.
- [23] R. Rahmatizadeh, P. Abolghasemi, A. Behal, and L. Bölöni, "From virtual demonstration to real-world manipulation using lstm and mdn," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.
- [25] 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.
- [26] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vaes: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations (ICLR)*, 2016.
- [27] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent service robotics*, 2016.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] H. G. Sung, "Gaussian mixture regression and classification," Ph.D. dissertation, RICE UNIVERSITY, 2004.
- [33] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*. M. Dekker New York, 1988.
- [34] M. Baucum, A. Khojandi, and T. Papamarkou, "Hidden markov models as recurrent neural networks: An application to alzheimer's disease," in *IEEE International Conference on Bioinformatics and Bioengineering (BIBE)*, 2021.
- [35] T. Hiraoka, S. Takase, K. Uchiumi, A. Keyaki, and N. Okazaki, "Recurrent neural hidden markov model for high-order transition," *Transactions on Asian and Low-Resource Language Information Processing*, 2021.
- [36] J. Chiu and A. M. Rush, "Scaling hidden markov language models," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.
- [37] A. K. Pandey and R. Gelin, "A mass-produced sociable humanoid robot: Pepper: The first machine of its kind," *IEEE Robotics & Automation Magazine*, 2018.
- [38] L. Fritsche, F. Unverzag, J. Peters, and R. Calandra, "First-person teleoperation of a humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015.
- [39] V. Prasad, R. Stock-Homburg, and J. Peters, "Learning human-like hand reaching for human-robot handshaking," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [40] 3DiVi, "Nuitrack." [Online]. Available: <https://nuitrack.com/>
- [41] A. Kshirsagar, R. Fortuna, Z. Xie, and G. Hoffman, "Dataset of bimanual human-to-human object handovers," *Data in Brief*, 2023.
- [42] E. Ng, Z. Liu, and M. Kennedy, "Diffusion co-policy for synergistic human-robot collaborative tasks," *IEEE Robotics and Automation Letters (RA-L)*, 2023.
- [43] K. Hansel, J. Urain, J. Peters, and G. Chalvatzaki, "Hierarchical policy blending as inference for reactive robot control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [44] S. Christen, W. Yang, C. Pérez-D'Arpino, O. Hilliges, D. Fox, and Y.-W. Chao, "Learning human-to-robot handovers from point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [45] P. Rosenberger, A. Cosgun, R. Newbury, J. Kwan, V. Ortenzi, P. Corke, and M. Grafinger, "Object-independent human-to-robot handovers using real time robotic vision," *IEEE Robotics and Automation Letters (RA-L)*, 2020.
- [46] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard, "A human-inspired controller for fluid human-robot handovers," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2016.
- [47] P. Khanna, M. Björkman, and C. Smith, "Human inspired grip-release technique for robot-human handovers," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2022.
- [48] A. Bolotnikoval, S. Courtois, and A. Kheddar, "Compliant robot motion regulated via proprioceptive sensor based contact observer," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2018.
- [49] M. Lagomarsino, M. Lorenzini, M. D. Constable, E. De Momi, C. Becchio, and A. Ajoudani, "Maximising efficiency of human-robot handovers through reinforcement learning," *IEEE Robotics and Automation Letters (RA-L)*, 2023.
- [50] M. Rubagotti, I. Tusseyeva, S. Baltabayeva, D. Summers, and A. Sandygulova, "Perceived safety in physical human-robot interaction—a survey," *Robotics and Autonomous Systems*, 2022.

## APPENDIX

### GAUSSIAN MIXTURE REGRESSION SIMPLIFICATION

Here, we simplify the covariance calculation in the GMR formulation (Eq. 5) to its equivalence to [27].

$$\begin{aligned}\hat{\Sigma}_i^r &= \Sigma_i^{rr} - \mathbf{K}_i \Sigma_i^{hr} + (\hat{\mu}_i^r - \hat{\mu}_t^r)(\hat{\mu}_i^r - \hat{\mu}_t^r)^T \\ &= \Sigma_i^{rr} - \mathbf{K}_i \Sigma_i^{hr} + \hat{\mu}_i^r (\hat{\mu}_i^r)^T - \hat{\mu}_i^r (\hat{\mu}_t^r)^T \\ &\quad - \hat{\mu}_t^r (\hat{\mu}_i^r)^T + \hat{\mu}_t^r (\hat{\mu}_t^r)^T \\ \hat{\Sigma}_t^r &= \sum_{i=1}^N \alpha_i(z_t^h) \hat{\Sigma}_i^r \\ &= \sum_{i=1}^N \alpha_i(z_t^h) (\Sigma_i^{rr} - \mathbf{K}_i \Sigma_i^{hr})\end{aligned}\tag{10}$$

$$\begin{aligned}&+ \sum_{i=1}^N \alpha_i(z_t^h) \hat{\mu}_i^r (\hat{\mu}_i^r)^T - \left( \sum_{i=1}^N \alpha_i(z_t^h) \hat{\mu}_i^r \right) (\hat{\mu}_t^r)^T \\ &- \hat{\mu}_t^r \sum_{i=1}^N \alpha_i(z_t^h) (\hat{\mu}_i^r)^T + \hat{\mu}_t^r (\hat{\mu}_t^r)^T \sum_{i=1}^N \alpha_i(z_t^h) \\ &= \sum_{i=1}^N \alpha_i(z_t^h) (\Sigma_i^{rr} - \mathbf{K}_i \Sigma_i^{hr} + \hat{\mu}_i^r (\hat{\mu}_i^r)^T) - \hat{\mu}_t^r (\hat{\mu}_t^r)^T\end{aligned}$$

since  $\sum_{i=1}^N \alpha_i(z_t^h) (\hat{\mu}_i^r) = \hat{\mu}_t^r$  and  $\sum_{i=1}^N \alpha_i(z_t^h) = 1$ , therefore  $-\hat{\mu}_t^r \sum_{i=1}^N \alpha_i(z_t^h) (\hat{\mu}_i^r)^T + \hat{\mu}_t^r (\hat{\mu}_t^r)^T \sum_{i=1}^N \alpha_i(z_t^h)$  becomes 0.