Transfer Learning of Shared Latent Spaces between Robots with Similar Kinematic Structure

Brian Delhaisse*†, Domingo Esteban*†, Leonel Rozo* and Darwin Caldwell*
*Department of Advanced Robotics
Istituto Italiano di Tecnologia
Via Morego 30, 16163 Genova, Italy
Email: name.surname@iit.it
†Both authors equally contributed to this paper.

***Abstract*—Learning complex manipulation tasks often requires to collect a large training dataset to obtain a model of a specific skill. This process may become laborious when dealing with high-DoF robots, and even more tiresome if the skill needs to be learned by multiple robots. In this paper, we investigate how this learning process can be accelerated by using shared latent variable models for knowledge transfer among similar robots in an imitation setting. For this purpose, we take advantage of a shared Gaussian process latent variable model to learn a common latent representation of robot skills. Such representation is then reused as prior information to train new robots by reducing the learning process to a latent-to-output mapping. We show that our framework exhibits faster training convergence and similar performance when compared to single- and multi-robot models. All experiments were conducted in simulation on three different robotic platforms: WALK-MAN, COMAN and CENTAURO robots.**

I. Introduction

Robot programming by demonstration (PbD) [1] is a compelling approach to teach robots new skills in a natural and user-friendly way that has been successfully applied for learning a large diversity of tasks [2], [3]. Nevertheless, it presents two challenges when it comes to teach both multiple and high DoF robots.

When the same set of skills needs to be learned by multiple robots it is necessary to teach each robot separately, increasing significantly the total training time and rendering the demonstration phase a tedious process. To alleviate this problem, the teacher can first train one robot and the acquired knowledge can be further used to train others, accelerating the overall training process. This scenario matches the paradigm of transfer learning, an approach where prior (learned) information is exploited in subsequent learning phases [4].

Moreover, when PbD is used for teaching skills to complex robots, like humanoids, the dimensionality of the training data can considerably increase, making both imitation and transfer learning more complex processes. A common choice to cope with high dimensional data is to first use dimensionality reduction techniques, and then, carry out the learning process.

Therefore, if we are interested in teaching the same set of skills to multiple humanoid robots from human demonstrations we have to address the two aforementioned problems. In this paper, we overcome the curse of dimensionality by assuming that both human and robot data lie in a shared low dimensional latent space. In this context, we exploit a shared Gaussian process latent variable model (GP-LVM) to learn this common representation [5], [6].

Moreover, if we consider that the human demonstrations have to produce similar movements in multiple robots, then the latent representation of one robot (source domain) can be used to learn a new shared GP-LVM for others (target domains). In other words, we propose that teaching a set of skills can be carried out by first learning a latent representation of the demonstrations, and then exploiting it for training other robots. By transferring this prior latent knowledge from one robot to another, we expect to achieve a faster learning rate along with good imitation performance (i.e., low reconstruction errors). Shared GP-LVM and the proposed extension for transfer learning are described in Section III.

The proposed approach was applied to imitate and reproduce bimanual movements of a human on three simulated robots, WALK-MAN, COMAN and CENTAURO. The details of the performed experiments and the analysis of their results are reported in Section IV. Limitations and potential extensions of our framework to kinematic-based hierarchical models and the application of reinforcement learning in the lower dimensional latent space are discussed in Section V.

II. Related Work

Teaching humanoids to execute a task from human demonstrations is a quite challenging task. This process is often carried out with many different sensory systems, such as RGBD cameras, wearable sensors, motion capture setups, among others, which generate a huge volume of data. One of the main problems when using these systems is to determine the kinematics correspondence between teacher and robot. In the existing literature, one can identify two prominent approaches to overcome this difficulty: based on kinematic analysis or built on machine learning models.

With respect to the former, for example, Yamane *et al.* proposed a control framework for simultaneous tracking and balancing of humanoids using human motion data [7]. A set of marker trajectories in Cartesian space describing the human motion was recorded, scaled to fit the robot size and, later, converted to joint trajectories through inverse kinematics. In [8], the authors defined the human motion as a sequence of postures, which were represented by Cartesian position of hands and feet measured with respect to the shoulder and hip frames. By considering a reference posture for both the human and the robot, relative positions of the limbs were translated to the robot using damped least-squares inverse kinematics.

A different approach based on virtual translational springs was introduced in [9]. The proposed method virtually connected the markers (placed on the human body) with corresponding points on the humanoid by using springs. The robot motion was guided through virtual spring forces, which were later used to generate motor torques on a simulated robot. This approach avoided the computation of an expensive inverse kinematics algorithm. Despite the foregoing works had significant results, they heavily depend on kinematic and human-designed models, which are hard to be applied when the teaching needs to be performed on multiple robots.

Machine learning approaches offer greater flexibility. For example, Stanton *et al.* proposed to learn a mapping between human and humanoid motions through feed-forward neural networks (NNs) defined for each robot joint. These NNs were trained with demonstration data obtained from a motion capture suit using particle swarm optimisation [10]. Although NNs are powerful to approximate non-linear functions, they typically require large training datasets, greatly prolonging the demonstration phase.

In [11], the authors proposed to learn a model of motion primitives for humanoid robots using hidden Markov models (HMM) and a mixture of Factor Analyzers (MFA). The main idea was to create a nonlinear low-dimensional mapping between human and robot using MFA, and reproduce the trajectories using an HMM-based dynamical system. Our work is similar to this approach in the sense that we also find a shared demonstrator-robot low dimensional manifold, but in our case the construction of the latent space and the mapping are wrapped in the same model, and therefore, their parameters are learned at the same time. Moreover, none of the aforementioned works addressed the problem of teaching skills to multiple robots. So, we go one step further by exploiting shared GP-LVM to deal not only with the curse of dimensionality and the human-robot mapping, but also with prior knowledge transfer - embedded in a latent representation of the motions - to quickly learn new mappings for multiple robots.

The use of dimensionality reduction techniques in transfer learning has also been proposed in [12]. This approach finds a common low-dimensional latent space between source and target domains (for input variables), and then learns a latent-to-output mapping in the source domain. The transfer learning occurs when this mapping is applied to the target domain. This approach differs from ours in that our latent space is not only between input and output spaces, but also between source and target domains. Moreover, our approach learns a new latent-to-output mapping in the target domain.

Transfer learning has also been applied to improve model learning in robotics. Nevertheless, these approaches do not make any distinction regarding the input and output variables of the problem, but they carry out a dimensionality reduction over the whole dataspace instead. By doing this, source and target low dimensional representations are different, and a linear [13] or a non-linear [14] transformation between them is required. In contrast, our framework does not group inputs and outputs together to perform the dimensionality reduction. In addition, no transformation between the models is required because the source and target inputs coincide, and therefore the same latent coordinates and the hyperparameters of the human-to-latent mapping are assumed to be shared among the robots.

Shared GP-LVMs have been successfully applied in computer graphics and robotics. Yamane *et al.* applied a shared GP-LVM to animate characters with different morphologies [15]. In robotics, Shon *et al.* applied this model for imitation of human actions in a small humanoid [5], [16]. Our approach extends on those works by considering multiple similar robots while differing in a major point: the input-to-latent mapping. In [15], given new inputs, the latent coordinates were determined using a nearest neighbor search and an optimization process. The performance heavily depends on the number of nearest points to be chosen, which may be considerably high, limiting the use of this approach for real-time systems. To alleviate this problem, Shon *et al.* applied a Gaussian process regression for the input-to-latent mapping once the model was trained. In our case, we impose back constraints between the input and the latent space, allowing to jointly learn the input-to-latent mapping along with the latent coordinates during the optimization. We explain the advantages of this decision in Section III-B, and experimentally demonstrate the benefits of imposing back constraints on the input in Section IV-B.

III. Proposed Approach

A. Notation and background

We start describing the notation and model that will be used to introduce the shared GP-LVM for transfer learning. First, let's denote N as the number of data points in our training set, $\boldsymbol{H} = [\boldsymbol{h}_1 \cdots \boldsymbol{h}_N]^T \in \mathbb{R}^{N \times D_H}$ as the human input data matrix with D_H representing the dimensionality of a human data point, and $\boldsymbol{R}^{(j)} = [\boldsymbol{r}_1^{(j)} \cdots \boldsymbol{r}_N^{(j)}]^T \in \mathbb{R}^{N \times D_{R^{(j)}}};\ \forall j \in \{1, ..., J\}$ as the j-th robot output data matrix where J denotes the total number of robots and $D_{R^{(j)}}$ the dimensionality of a robot data point. Also, let us define $\boldsymbol{X} \in \mathbb{R}^{N \times D_X}$ as the latent matrix (with dimension $D_X < \min(D_H, D_R)$), $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ as a kernel (a.k.a covariance) matrix, $\boldsymbol{\Phi}$ as the hyperparameters of the kernel, and $\boldsymbol{Y} \in \{\boldsymbol{H}, \boldsymbol{R}\}$ to represent the observed human or robot data.

Gaussian Processes (GP) define probability distributions over functions in which any finite collection of the function values are assumed to be jointly Gaussian [17]. In addition to be non-parametric, their attractiveness is empowered by their computational tractability for inference and learning. This general framework works by placing a Gaussian prior over functions: $\boldsymbol{f}|\boldsymbol{H} \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}(\boldsymbol{H}, \boldsymbol{H}'))$, with zero mean and covariance matrix characterized by the kernel $\boldsymbol{K}$. The kernel allows us to encode our prior beliefs about the set of plausible functions. A popular one is the Squared Exponential (SE) kernel[1] which is infinitely differentiable and appropriate for modelling smooth functions. Extended with Automatic Relevance Determination (ARD) [17], each entry of the kernel matrix is given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \alpha^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{2l_d}\right) + \beta\delta_{\boldsymbol{x},\boldsymbol{x}'} \quad (1)$$

in which the amplitude α, the lengthscales l_d, and the variance β of the white noise term are the kernel hyperparameters (denoted by $\boldsymbol{\Phi}$) that will be optimized.

By observing some pairs of data points $\{(\boldsymbol{h}_i, \boldsymbol{r}_i)\}_{i=1}^N$, we can update our beliefs about the distribution of functions. Assuming the likelihood is also Gaussian, that is, $p(\boldsymbol{R}|\boldsymbol{F}) = \prod_{n=1}^N \mathcal{N}(\boldsymbol{r}_n|\boldsymbol{f}_n, \sigma^2\boldsymbol{I})$, where σ^2 represents the noise variance and $\boldsymbol{F} = [\boldsymbol{f}_1, ..., \boldsymbol{f}_N]^T$, the marginal likelihood expressed as

$$p(\boldsymbol{R}|\boldsymbol{H}; \boldsymbol{\Phi}) = \int p(\boldsymbol{R}|\boldsymbol{F})p(\boldsymbol{F}|\boldsymbol{H}; \boldsymbol{\Phi})d\boldsymbol{F}, \quad (2)$$

can then be computed analytically. The maximization of (2) permits to find the optimal set of hyperparameters $\boldsymbol{\Phi}^*$ for the GP, that is,

$$\boldsymbol{\Phi}^* = \underset{\boldsymbol{\Phi}}{\operatorname{argmax}}\, p(\boldsymbol{R}|\boldsymbol{H}; \boldsymbol{\Phi}). \quad (3)$$

While Gaussian process regression gives us a direct mapping from the input $\boldsymbol{H}$ to the output $\boldsymbol{R}$, it does not provide a model in which knowledge can be easily exploitable and transferable to other robots. To address this issue, we make use of shared latent spaces. We first begin by reviewing how latent variable models can be represented using Gaussian processes.

The Gaussian process latent variable model (GP-LVM) is a non-parametric probabilistic model which performs a non-linear dimensionality reduction over observed data [18]. In this model, marginalization is carried out over the parameters instead of the latent variables, which are optimized. Specifically, the marginal likelihood for $\boldsymbol{Y}$ given the latent coordinates $\boldsymbol{X}$ is specified by:

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{X}; \boldsymbol{\Phi}_Y) &= \prod_{d=1}^{D_Y} \mathcal{N}(\boldsymbol{Y}_{:,d}|\boldsymbol{0}, \boldsymbol{K}_Y) \\ &= \frac{1}{\sqrt{(2\pi)^{ND_Y}|\boldsymbol{K}_Y|^{D_Y}}} \exp\left(-\frac{1}{2}tr(\boldsymbol{K}_Y^{-1}\boldsymbol{Y}\boldsymbol{Y}^T)\right) \end{aligned}$$

where $\boldsymbol{Y}_{:,d}$ denotes the d-th column of $\boldsymbol{Y}$. The maximization of this term not only provides us the optimal hyperparameters, but also the optimal set of latent points, that is,

$$\{\boldsymbol{X}^*, \boldsymbol{\Phi}_Y^*\} = \underset{\boldsymbol{X}, \boldsymbol{\Phi}_Y}{\operatorname{argmax}}\, p(\boldsymbol{Y}|\boldsymbol{X}; \boldsymbol{\Phi}_Y). \quad (4)$$

While this unsupervised model provides us the latent coordinates, it does not account for the human and the robot(s) data simultaneously.

To conform with the previous requirement, we make use of an extended version of the GP-LVM, called shared GP-LVM. In this setting, multiple observational spaces share a common latent space [6] (see Figs. 1 and 2). With k observational spaces, the marginal likelihood to maximize is then given by

$$\{\boldsymbol{X}^*, \{\boldsymbol{\Phi}_{Y^{(i)}}^*\}_{i=1}^k\} = \underset{\boldsymbol{X}, \{\boldsymbol{\Phi}_{Y^{(i)}}\}_{i=1}^k}{\operatorname{argmax}}\, p(\{\boldsymbol{Y}^{(i)}\}_{i=1}^k|\boldsymbol{X}; \{\boldsymbol{\Phi}_{Y^{(i)}}\}_{i=1}^k), \quad (5)$$

where the hyperparameters $\boldsymbol{\Phi}$ of each kernel, and the latent coordinates $\boldsymbol{X}$ are jointly optimized.

In order to preserve local distances and have a smooth mapping from one of the observational spaces to the latent space, back constraints are introduced. This feature is useful as in the teleoperation case we are seeking for similar inputs to be mapped to similar outputs. With back constraints placed on one of the those spaces [19], the objective function to maximize becomes

$$\{\boldsymbol{W}^*, \{\boldsymbol{\Phi}_{Y^{(i)}}^*\}_{i=1}^k\} = \underset{\boldsymbol{W}, \{\boldsymbol{\Phi}_{Y^{(i)}}\}_{i=1}^k}{\operatorname{argmax}}\, p(\{\boldsymbol{Y}^{(i)}\}_{i=1}^k|\boldsymbol{W}; \{\boldsymbol{\Phi}_{Y^{(i)}}\}_{i=1}^k), \quad (6)$$

with $\boldsymbol{X} = g(\boldsymbol{Y}^{(i)}; \boldsymbol{W})$ where g is a function parametrized by weights $\boldsymbol{W}$, which are learned during the optimization process. By placing back constraints either on the input or the output, this results in different performances and behaviors which will be further described in section IV-B.

Once the model has been trained, given new inputs $\hat{\boldsymbol{H}}$, predictions in the shared GP-LVM are realized by first projecting the human input data to the latent space, and then projecting it to the robot output space. In the case back constraints are placed on the input space, the input-to-latent mapping is performed using the learned parametric function $g(\hat{\boldsymbol{H}}; \boldsymbol{W})$. Alternatively, this mapping might be achieved by carrying out a GP regression once the model has been learned, as performed in [5]. While other approaches for this mapping are available, they do not fulfill our real-time requirements in order to teleoperate the robot, in the sense that they usually employ some optimization process for each prediction [6], [15].

B. Shared GP-LVM for Transfer Learning

Exploiting the latent coordinates to transfer acquired knowledge to other robots is a promising approach. This has the potential to lead to faster learning and similar reproduction performance compared to their fully trained counter parts.

In this context, we define a shared GP-LVM which is first fully trained on a specific robot, and then transferred to other robots with partial re-training, as shown in Fig. 1. Assuming back constraints on the input, we first fully train a shared GP-LVM model on one particular robot by maximizing the following marginal likelihood:

$$\{\boldsymbol{W}^*, \boldsymbol{\Phi}_H^*, \boldsymbol{\Phi}_{R^{(1)}}^*\} = \underset{\boldsymbol{W}, \boldsymbol{\Phi}_H, \boldsymbol{\Phi}_{R^{(1)}}}{\operatorname{argmax}}\, p(\boldsymbol{H}, \boldsymbol{R}^{(1)}|\boldsymbol{W}; \boldsymbol{\Phi}_H, \boldsymbol{\Phi}_{R^{(1)}}). \quad (7)$$

This allows us to obtain the optimal latent coordinates $\boldsymbol{X}^* = h(\boldsymbol{H}; \boldsymbol{W}^*)$, and optimal hyperparameters for the mappings. This pretrained model is then transferred to another robot which involves its latent-to-output mapping to be optimized while the latent points and input-to-latent mapping are

[1] also known as Radial Basis Function (RBF) kernel or Gaussian kernel.

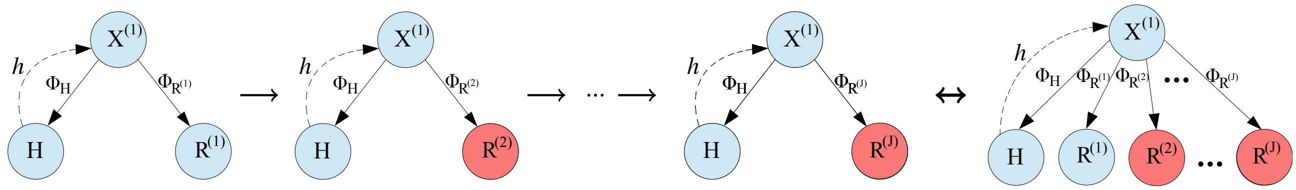


Fig. 1: A shared GP-LVM fully trained on one robot $\boldsymbol{R}^{(1)}$, that is, the hyperparameters $\boldsymbol{\Phi}_H$, $\boldsymbol{\Phi}_{R^{(1)}}$, and the latent coordinates $\boldsymbol{X}^{(1)}$ are jointly optimized. This model is then transferred to another robot $\boldsymbol{R}^{(2)}$ in which the latent coordinates and the hyperparameters $\boldsymbol{\Phi}_H$ are maintained fixed while the hyperparameters $\boldsymbol{\Phi}_{R^{(2)}}$ for the new latent-to-output mapping are optimized. This process is carried out over all the other robots $\boldsymbol{R}^{(j)}$, $\forall j \in \{3, ..., J\}$. The equivalent system of the whole process is depicted on the rightmost figure. Once the optimization process is over, new human input data are given to the system which produce the corresponding output data for each robot.

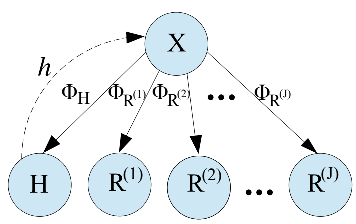


Fig. 2: In this multi-robot learning model, the latent coordinates and the hyperparameters of each robot are jointly optimized. This leads to a model which is less biased to a specific initial selected robot, as it needs to compromise between all the robots.

maintained fixed. This optimization process is carried out for any other robot $j \in \{2, ..., J\}$,

$$\boldsymbol{\Phi}^*_{R^{(j)}} = \underset{\boldsymbol{\Phi}_{R^{(j)}}}{\operatorname{argmax}}\, p(\boldsymbol{H}, \boldsymbol{R}^{(j)} | \boldsymbol{X}^*; \boldsymbol{\Phi}^*_H, \boldsymbol{\Phi}_{R^{(j)}}). \tag{8}$$

As only the latent-to-output mapping is optimized here, this naturally leads to a faster learning process. This model is equivalent to a shared GP-LVM in which all the other robots have been appended to the initial fully trained model (see Fig. 1 rightmost picture).

In contrast to our approach built on transfer learning, we also define an alternative model that will be employed in our experiments. This model describes a multi-robot system in which all the human and robot spaces are generated by a common latent space (see Fig. 2). In this framework, all the hyperparameters as well as the latent coordinates are being jointly optimized, in constrast to the model shown in Fig. 1, where prior information is exploited. The optimization process has to make a tradeoff between the latent coordinates and the hyperparameters of each kernel for the human and all the robots. This procedure is specified by

$$\left\{ \begin{matrix} \boldsymbol{W}^*, \boldsymbol{\Phi}^*_H, \\ \{\boldsymbol{\Phi}^*_{R^{(i)}}\}_1^J \end{matrix} \right\} = \underset{\boldsymbol{W}, \boldsymbol{\Phi}_H, \{\boldsymbol{\Phi}_{R^{(i)}}\}_1^J}{\operatorname{argmax}} p(\boldsymbol{H}, \{\boldsymbol{R}^{(i)}\}_1^J | \boldsymbol{W}; \boldsymbol{\Phi}_H, \{\boldsymbol{\Phi}_{R^{(i)}}\}_1^J). \tag{9}$$

In order for the transfer to be successful, we impose a back-constraint on the input motivated by four reasons. Firstly, we are interested in having a smooth input-to-output mapping. While a GP-LVM already provides a smooth mapping from the latent space to the observational spaces, the converse is not true [19]. Thus, in our case, we also seek to have a smooth mapping from the input to the latent space. Secondly, constraining the latent space (and thus the latent parameters) by any means would be beneficial as it would (at the expense of an introduced bias) decrease the variance and thus make it more robust to overfitting. Thirdly, when transfering the model to another robot, the human input data is fixed while the robot output data changes. Imposing back constraints on the input therefore renders it less dependent on the output, and thus more stable and robust. Finally, we exploit the ability to jointly optimize the input-to-latent mapping along with the model. The latent coordinates are hence undirectly optimized while being still dependent on the back constraint function. This is relevant as this mapping will be used later when predicting the latent coordinates given the new input data.

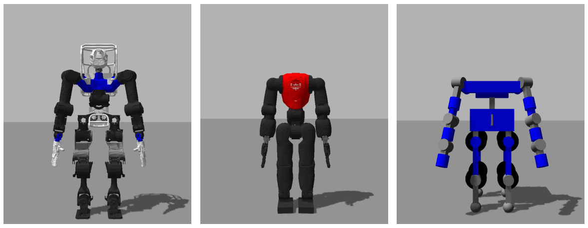
As previously introduced, the function h (see Figs. 1 and 2) represents the inverse mapping, which given the human input data outputs the latent coordinates. As described in section III-A, this function can be learned during optimization as it happens when back constraints are imposed on the input (i.e. $h \equiv g$), or after by applying a GP regression from the input to the latent space. In section IV-B, we will experimentally show that imposing back constraints on the input does not only greatly improve the performances of the fully trained model, but is indispensable for the transfer to be successful.

IV. Experiments

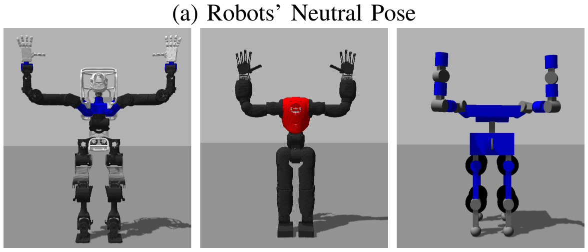
A. Setup Description

The proposed framework was evaluated reproducing bimanual movements of a human in three simulated robots WALK-MAN [20], COMAN [21] and CENTAURO [22], spawned in Gazebo simulator. Bimanual skills are of high interest because they enrich the robot dexterity, allowing it to perform a larger set of manipulation tasks [23].

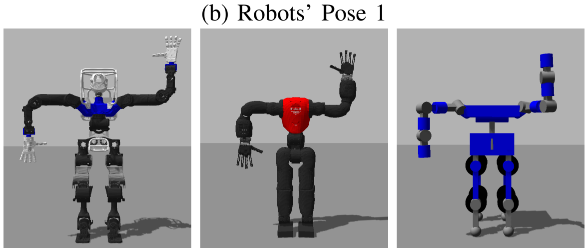
The process used to collect the human and robots data began by defining a set of 16 key bimanual poses for each robot, as depicted in Fig. 3. Then, robot trajectories between these poses



(a) Robots' Neutral Pose



(b) Robots' Pose 1



(c) Robots' Pose 3

Fig. 3: Three of the sixteen robots' poses defined for each robot. From left to right: WALK-MAN, COMAN and CENTAURO.

were generated using a quintic Hermite interpolator with zero initial and final velocities and accelerations. Next, a human, wearing a Xsens MVN BIOMECH suit, mimicked the robots' motion. To collect variability in the data, the human went back and forth between different poses three times. During this process, the joint values of the human and the desired joint commands of the robot were recorded at 40 Hz.

Once the data was collected, all the trajectories starting from the neutral pose were grouped as training set, and the trajectories between different poses excluding the neutral pose as test set. Finally, we subsampled the trajectories categorized as training data with the objective of reducing the number of data points. The data retrieved by the motion capture system had 9 DoF for each arm, so the human data dimensionality $D_H = 18$. On the other hand, the three simulated robots have 7 revolute joints in each arm, thus $D_{R^{(1)}} = D_{R^{(2)}} = D_{R^{(3)}} = 14$.

The collected dataset were used in our experiments to (i) corroborate our hypothesis that shared latent spaces can be exploited to transfer prior information, (ii) analyze the benefits of applying back constraints in a shared GP-LVM between human and robot, (iii) compare the multi-robot learning models described in Section III-B with independent shared GP-LVM for each robot. These models were implemented, trained and tested by extending the GPy framework [24].

Three shared GP-LVMs were modeled to analyze the benefits of back constraints. The first model did not have back constraints and the second model presented back constraints from the robot joints, $\mathbf{R}^{(j)}$, to the latent space, $\mathbf{X}$. In both, the mapping from human data $\mathbf{H}$ to $\mathbf{X}$ was carried out with a Gaussian process regression, once the shared model learned. The third model had back constraints from $\mathbf{H}$ to $\mathbf{X}$ so the corresponding mapping was performed by the learned parametric function $g(\hat{\mathbf{H}}; \mathbf{W})$, which in our case was a RBF kernel based mapping. The latent locations in the shared GP-LVM without back constraints were initialized by averaging the PCA solutions of $\mathbf{H}$ and $\mathbf{R}^{(j)}$. This initialization was not required in the models using back constraints where the latent locations are parametrized by $\mathbf{W}$.

The shared GP-LVM shown in Fig. 1 was modeled by replacing the corresponding joint angles of one robot by another, and training only the hyperparameters of the latent-to-output mapping. On the other hand, the multi-robot model depicted in Fig. 2 was characterized by a multi-output structure containing all the joint angles of the three robots. In all the shared GP-LVMs, we set the dimension of the shared latent space to $D_X = 5$, a value that reduced the dimension of the latent space while still achieving a performance similar to a GP regression representing a direct human-robot mapping. The SE kernel function with ARD was used, and the hyperparameters along with the latent positions (in the case without back constraints) were optimized with the L-BFGS-B algorithm.

B. Results

To corroborate our hypothesis that shared latent spaces can be exploited to transfer prior knowledge, we first tested our proposed model (described in section III-B and depicted in Fig. 1) without back constraints. Results for each robot are reported in Fig. 4. The plots show the reconstruction errors on a test dataset corresponding to the proposed shared GP-LVM for transfer learning, a model fully trained on a single robot dataset (used as baseline), and the alternative multi-robot model shown in Fig. 2. Both the baseline and multi-robot models were averaged over 10 runs while the proposed model was averaged over 10 runs for each run of the fully trained model, thus a total of 100 runs. By comparing the three plots, we observe that the MSE achieved by the baseline model on one robot is similar to the performance of our model pretrained on the same robot. This observation is consistent with our expectation that the optimization of the latent-to-output mapping of our pretrained models should not influence excessively the results, as the latent coordinates and input-to-latent mapping remain unchanged. While we indeed observe faster convergence for our proposed shared GP-LVM, because of the smaller amount of hyperparameters to optimize, a similar test error is not observed between the different models. To address this issue, we incorporated back constraints into our model.

For the various reasons mentioned in section III-B, we apply back constraints on the input in our model. Results of imposing such constraint for each robot and each model are reported in Fig. 5. In contrast to the results shown in Fig. 4,

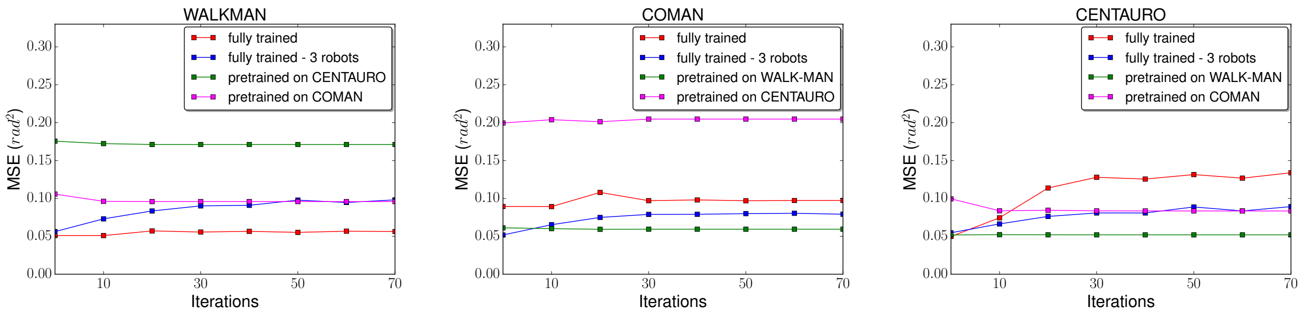


Fig. 4: Mean squared error of the pretrained and fully trained models with no back constraints for the three robots on the test dataset.

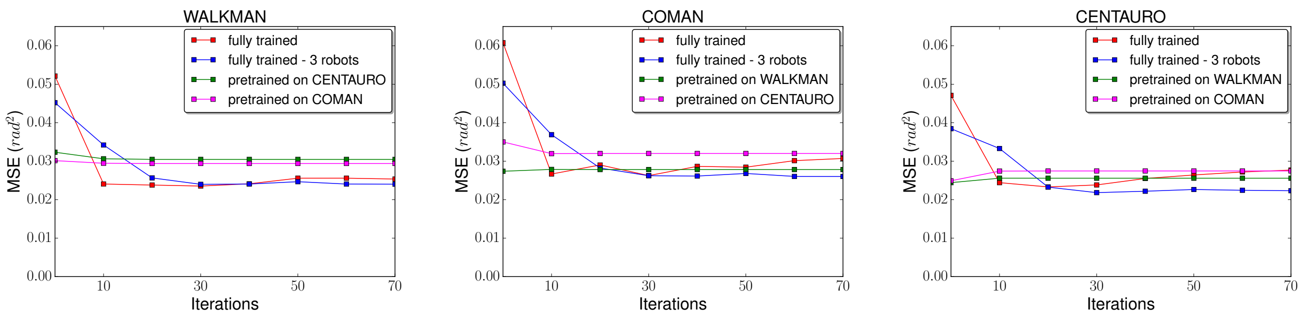


Fig. 5: Mean squared error of the pretrained and fully trained models with back constraints set on the input for the three robots on the test dataset.

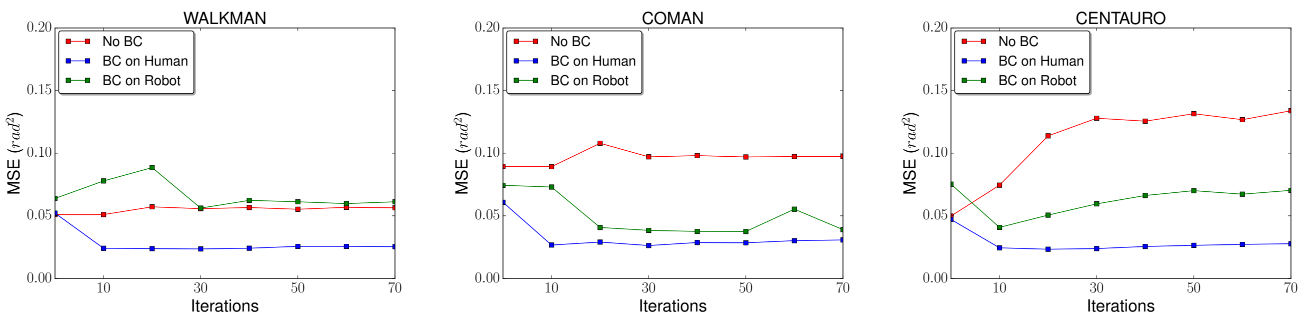


Fig. 6: Mean squared error showing the performance without back constraints (red), with back constraints placed on the input (blue) or on the output (green) for the three robots on the test dataset.

the obtained results are significantly better, and are similar between the different models. Moreover, we still observe a faster convergence for our proposed model. This faster convergence rate is attractive especially in the presence of larger datasets. To further confirm the benefits of using back constraints, we performed an additional test. We compared the fully trained model on each robot with back constraints on the input, output, and without back constraints (see Fig. 6). The plots validate our assumption that back constraints in general are beneficial compared to the non back constraints case. More importantly, they show that imposing back constraints on the input results in the best performance corroborating our choice.

Having confirmed that placing back constraints on the input is beneficial for the transfer to be successful, we now inspect a particular instance with the model trained on WALK-MAN and then transferred to COMAN. We are mainly interested by the structure of the latent space and the predictions made by our proposed model given new human input data. The learned shared latent space for the WALK-MAN robot with back constraints on the input is depicted in Fig. 7. The represented poses are the same as those shown in Fig. 3. The lines represent the resulting latent trajectories for some of the human test data. While these trajectories show clearly the forth and back movement, they are not smooth because of the noisy human input data.

The predictions made by the model given the trajectories of

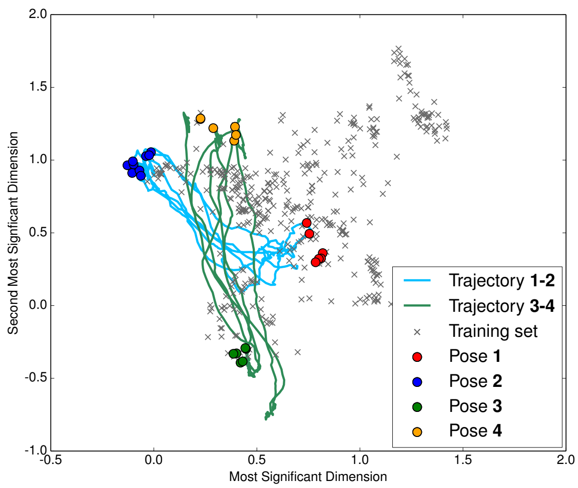


Fig. 7: Shared latent space of a human and the WALK-MAN robot with back constraints on the input. The circles represent four of the predefined key poses, while the crosses represent the positions during the training movements.

the human arms when moving back and forth between the pose 1 and 2 are reported in Fig. 8. This corresponds to the blue trajectory in the latent space shown in Fig. 7. The pose 1 is depicted in Fig. 3b while the pose 2 is similar to the former one but with the shoulders rotated $180°$ in the sagittal plane. The model was first fully trained on WALK-MAN, then transferred to COMAN where the latent-to-output mapping was learned. It can be seen that the predicted robot joint trajectories follow the desired pattern, are robust to a certain extent to noisy input data, and can deal satisfactorily with the difference between the human and robot kinematic structure.

V. Discussion

A. Challenges

We have experimentally shown that our proposed extension of shared GP-LVM for transfer learning between similar robots is a promising approach. This model offers faster convergence rates, and robot motions that replicate the input human poses. However, some challenges need to be addressed to further exploit the potential of this work.

The first open issue concerns the latent space dimensionality, which is currently set by the user. In our framework, the latent dimension was set to 5 motivated by our desire to reduce as much as possible the dimensionality of the latent space while still achieving good performances close to a GP regression. Choosing a specific dimensionality can be cumbersome as it requires testing different latent dimensions to satisfy different criteria. We may overcome this limitation by using variational bayesian techniques to automatically estimate the dimensionality of the shared GP-LVM.

Our current framework focuses on robots which have similar kinematic structure, and the model was experimentally tested using only bimanual gestures. Robots with significant differences in their kinematics should be further investigated. However, in view of our obtained results, we postulate that our proposed model with back constraints on the input will still lead to low reconstruction errors even in the case of different kinematic structures. The reason being that the latent coordinates will be constrained by the human input data which, in contrast to the robot output data, do not vary. This stronger dependence between the human input and latent space makes it less sensible to the robot output data.

Additionally, we carried out some experiments to demonstrate that the proposed transfer learning process also allows us to train the transferred shared GP-LVMs with less datapoints, and obtain better results than the models that were indepently trained for each robot. Depending on the mapping function used for back constraints, there exists some evidence suggesting this benefit, because of lower MSE in the pretrained models, but the results were not conclusive and require more extensive experimentation.

B. Future work

As mentioned in [16], having separate shared GP-LVMs for each body part leads to better results when independent movements need to be produced. Moreover, hierarchical formulations encode a tree-based model in the latent variables for each body part, allowing for more complex hierarchies to transition or interpolate between basis behaviors [25]. In this context, we may consider that a source shared latent graph can also be transferred to other robots in a similar way than our approach.

The shared latent space can be further exploited to refine acquired skills through the use of reinforcement learning, which can benefit from the lower dimensionality provided by this model to obtain faster refinements. Policies learned and parametrized in the latent space may potentially be shared with multiple robots.

VI. Conclusion

In this paper, we extended a shared GP-LVM to transfer knowledge between robots with similar kinematic structure. We demonstrated that a latent space shared between a human and a robot can be transferred to others for the reproduction of bimanual movements. As a result, we obtained faster convergence in the training process compared to single models that were indepently trained for each robot. In our approach, transfer learning is possible because of the application of back constraints between the human and the latent space when optimizing the model. This results in smooth transitions from the input to the latent representation and avoids the need to train these parameters for new robots. The proposed approach may be nicely exploited to teach multiple similar robots a task from teleoperated demonstrations using an unique model.

References

[1] A. Billard, S. Calinon, and R. Dillmann, "Learning from humans," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2016, ch. 74, pp. 1995–2014, 2nd Edition.

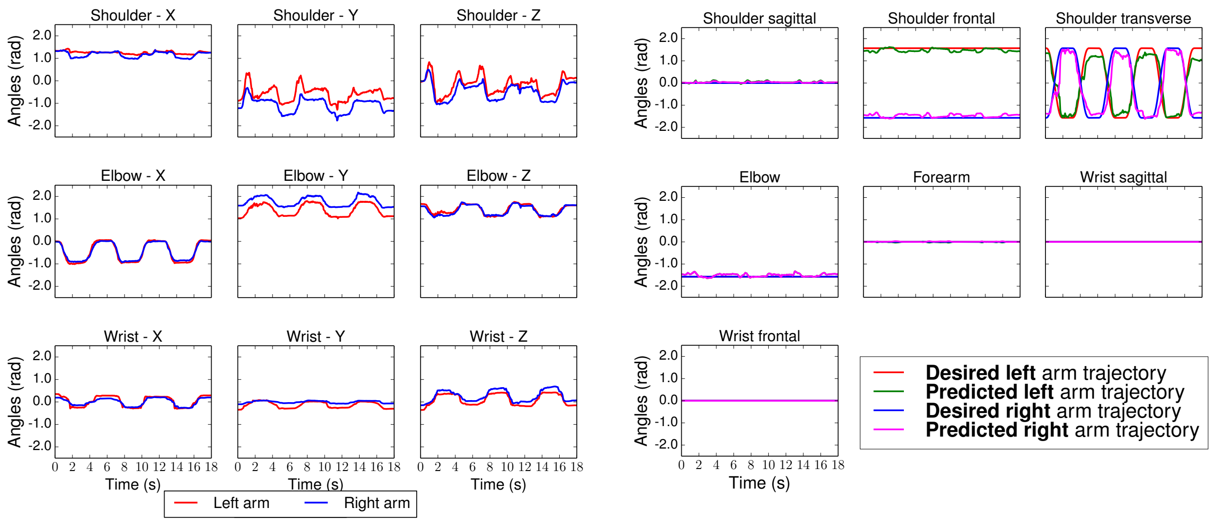


Fig. 8: Prediction plots for COMAN: Human left and right arm trajectories (left), and corresponding predicted left and right robot arm trajectories (right). A shared GP-LVM is trained on WALK-MAN with back constraints set on the input, then transferred and partially trained on COMAN. The robot trajectories are the movements performed by COMAN which result from this transfer learning. The human trajectories are part of the test dataset, and the corresponding latent trajectories are depicted in blue in Fig. 7.

[2] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *IEEE Intl. Conf. on Robotics and Automation*, 2009, pp. 763–768.

[3] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Trans. on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[5] A. P. Shon, K. Grochow, A. Hertzmann, and R. P. Rao, "Learning shared latent structure for image synthesis and robotic imitation," in *Advances in Neural Information Processing Systems*, 2006, pp. 129–134.

[6] C. H. Ek, "Shared gaussian process latent variables models," Ph.D. dissertation, Oxford Brookes University, 2009.

[7] K. Yamane, Y. Ariki, and H. Jessica, "Controlling humanoid robots with human motion data: Experimental validation," in *IEEE-RAS Intl. Conf. on Humanoid Robots*, 2010, pp. 504–501.

[8] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *IEEE Intl. Conf. on Robotics and Automation*, 2014, pp. 2806–2812.

[9] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human-humanoid interaction," *Intl. Journal of Robotics Research*, vol. 29, no. 13, pp. 1684–1704, 2010.

[10] C. Stanton, A. Bogdanovych, and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning," in *Australasian Conf. on Robotics and Automation*, 2012.

[11] M. Field, D. Stirling, Z. Pan, and F. Naghdy, "Learning trajectories for robot programing by demonstration using a coordinated mixture of factor analyzers," *IEEE Trans. on Cybernetics*, vol. 46, no. 3, pp. 706–717, 2016.

[12] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *AAAI Conf. on Artificial Intelligence*, 2008, pp. 677–682.

[13] B. Bócsi, L. Csató, and J. Peters, "Alignment-based transfer learning for robot models," in *Intl. Joint Conf. on Neural Networks*, 2013, pp. 1–7.

[14] N. Makondo, B. Rosman, and O. Hasegawa, "Knowledge transfer for learning robot models via local procrustes analysis," in *IEEE-RAS Intl. Conf. on Humanoid Robots*, 2015, pp. 1075–1082.

[15] K. Yamane, Y. Ariki, and H. Jessica, "Animating non-humanoid characters with human motion data," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010, pp. 169–178.

[16] A. P. Shon, K. Grochow, and R. P. Rao, "Robotic imitation from human motion capture using gaussian processes," in *IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 129–134.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[18] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in Neural Information Processing Systems*, 2004, pp. 329–336.

[19] N. D. Lawrence and J. Quiñonero-Candela, "Local distance preservation in the gp-lvm through back constraints," in *Intl. Conf. on Machine learning*, 2006, pp. 513–520.

[20] N. G. Tsagarakis, D. G. Caldwell, and et al., "Walk-man: a high performance humanoid platform for realistic environments," *Journal of Field Robotics*, 2016.

[21] N. G. Tsagarakis, S. Morfey, G. M. Cerda, L. Zhibin, and D. G. Caldwell, "Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control," in *IEEE Intl. Conf. on Robotics and Automation*, 2013, pp. 673–678.

[22] M. Kameduła, N. Kashiri, D. G. Caldwell, and N. G. Tsagarakis, "A compliant actuation dynamics gazebo-ros plugin for effective simulation of soft robotics systems: Application to centauro robot," in *Intl. Conf. on Informatics in Control, Automation and Robotics*, 2016, pp. 485–491.

[23] J. Silvério, L. Rozo, S. Calinon, and D. G. Caldwell, "Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems," in *IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[24] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, since 2012.

[25] N. D. Lawrence and A. J. Moore, "Hierarchical gaussian process latent variable models," in *Intl. Conf. on Machine Learning*, 2007, pp. 481–488.