International Journal of Humanoid Robotics
Vol. 17, No. 1 (2020) 2050001 (23 pages)

DOI: 10.1142/S0219843620500012

Incremental Learning of an Open-Ended Collaborative Skill Library

Dorothea Koert[*,§], Susanne Trick[†], Marco Ewerton[*],
Michael Lutter[*] and Jan Peters[*,‡]

[*]*Intelligent Autonomous Systems Group,*
Department of Computer Science,
Technische Universität Darmstadt, Darmstadt, Germany

[†]*Psychology of Information Processing Group,*
Department of Psychology,
Technische Universität Darmstadt, Darmstadt, Germany

[‡]*Max Planck Institute for Intelligent Systems,*
Tübingen, Germany
[§]*koert@ias.tu-darmstadt.de*

Received 30 March 2019
Revised 25 October 2019
Accepted 20 November 2019
Published 10 January 2020

Intelligent assistive robots can potentially contribute to maintaining an elderly person's independence by supporting everyday life activities. However, the number of different and personalized activities to be supported renders pre-programming of all respective robot behaviors prohibitively difficult. Instead, to cope with a continuous and potentially open-ended stream of cooperative tasks, new collaborative robot behaviors need to be continuously learned and updated from demonstrations. To this end, we introduce an online learning method to incrementally build a cooperative skill library of probabilistic interaction primitives. The resulting model chooses a corresponding robot response to a human movement where the human intention is extracted from previously demonstrated movements. While existing batch learning methods for movement primitives usually learn such skill libraries only once for a pre-defined number of different skills, our approach enables extending the skill library in an open-ended and online fashion from new incoming demonstrations. The proposed approach is evaluated on a low-dimensional benchmark task and in a collaborative scenario with a 7DoF robot, where we also investigate the generalization of learned skills between different subjects.

Keywords: Learning from demonstrations; human robot interaction.

1. Introduction

The expected demographic change is an urgent and prevailing challenge for society. An increasing number of elderly people need assistance in their daily lives while only few caregivers are available.[1] In order to address this challenge, the development of

[§]Corresponding author.

technical solutions for elderly assistance is essential. In particular, assistive robots can potentially support elderly people in their daily lives and thus help to maintain their personal independence. Such cooperative robot assistants need the ability to support multiple different tasks and additionally should be able to individually adapt to a user's needs. This disqualifies pre-programming of all possible tasks. Instead, an intuitive way to teach multiple personalized skills to a robot is needed, which can be realized by Learning from Demonstrations (LfDs).[2]

Since human motions exhibit a high variability,[3] a probabilistic approach for cooperative skill LfDs is required, which takes variations in human motions during demonstrations and execution time into account. In addition, to be able to learn cooperative skills from demonstrations and representing them probabilistically, it is not just crucial to detect multiple different skills but particularly to update an existing skill library with completely new cooperative skills and refining the existing skills with new demonstrations.

This paper presents a novel online learning method for building a collaborative skill library, which enables open-ended learning of new skills and refinement of existing skills. Figure 1 summarizes our approach. An incremental mixture model of probabilistic movement primitives is proposed for online learning of collaborative skills from demonstrations. Cooperative skills are represented as probabilistic interaction movement primitives (Interaction ProMPs)[4] which capture correlations between human and robot movements as well as the inherent variance. Although these movement primitives have already been used for learning multiple different cooperative skills,[5] to the best of our knowledge, none of the present approaches is able to iteratively learn multiple Interaction ProMPs online and in an open-ended

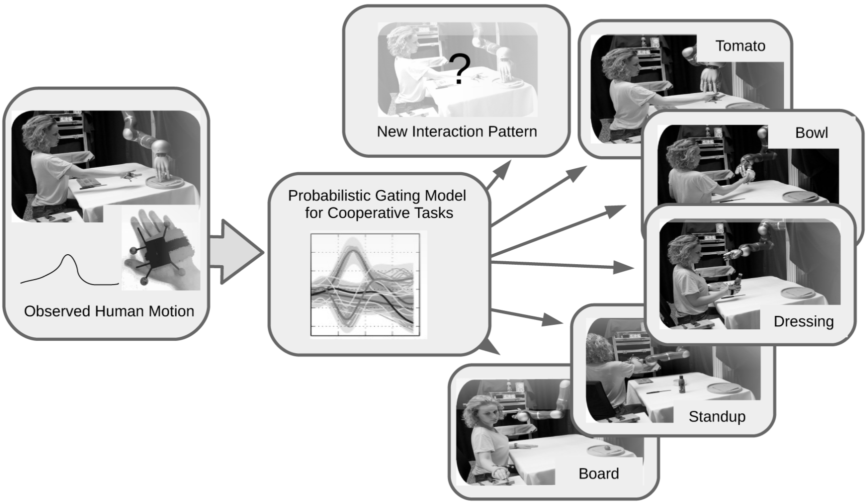


Fig. 1. Intelligent assistive robots should learn multiple cooperative tasks from a continuous and open-ended stream of new demonstrations. To this end, we propose a novel approach for incremental and open-ended learning of a mixture model of probabilistic interaction primitives.

way. However, for personalized robot assistants it is crucial to open-endedly learn new tasks and continuously update existing cooperative skills with new demonstrations. In particular, in such an open-ended scenario the total number of cooperative tasks cannot be known beforehand and thus needs to be extended during the learning process. In contrast to prior work on learning a Mixture of Interaction primitives,[5] our new approach does not rely solely on demonstrations which are available at the first training time but can integrate new demonstrations and tasks over multiple training sessions. In addition to the experimental validation previously reported in Ref. 6, we extended our method to learn a library across multiple subjects. To enable successful incremental learning of a skill library on data of multiple subjects we introduce ageing out and consolidation of model components and an automated normalization routine based on personal workspace and start position of the subjects.

The rest of this paper is structured as follows: Section 2 discusses related work. Section 3 provides an overview on the existing approach of Batch Learning for a Mixture of Interaction Primitives and introduces our novel approach for Online Open-Ended Learning of a Mixture of Interaction Primitives. In Sec. 4, we evaluate this new approach on 2D trajectory data and in a collaborative robotic scenario, where we additionally investigate how the learned skills transfer between different subjects. Finally, we conclude with Sec. 5 and discuss ideas for future work.

2. Related Work

Learning cooperative tasks between humans and robots from demonstration is a popular approach as it enables also non-expert users to teach personalized skills to robots.[7–11] While a number of existing learning from demonstration literature focusses on task representation for a single agent for collaborative tasks it is also desirable to address coupled skill representations, e.g. for human and robot motions.[12–14] In particular, the concept of movement primitives offers hereby a lower dimensional representation of trajectories and a modular framework that does not only reproduce demonstrated behaviors but potentially also generalizes to new situations[10,12,15–18] In order to capture also variability in non-deterministic human motion demonstrations it is desirable to not only model demonstrations with single trajectories[15] but to also capture the variability using a probabilistic approach and distributions over trajectories.[10,16,17] Probabilistic Interaction Movement Primitives (Interaction ProMPs)[4,11,19] offer hereby a probabilistic representation to model inherent correlations in the movements of two actors, such as human and robot, from coupled demonstrated trajectories while showing beneficial interpolation capabilities compared to other movement primitive approaches.[17]

However, to achieve a personalized cooperative robot it is desirable to learn multiple cooperative tasks and decide on their activation depending on the context or on the human intention.[20–22] To this end, an approach that deploys Gaussian Mixture Models (GMMs) and Expectation Maximization (EM) to learn multiple Interaction ProMPs from unlabeled demonstrations has been introduced.[5] This approach considers batch data, i.e. assuming the availability of all data during training. This limits its application to settings where the number of tasks does not change after training and no new demonstration trajectories need to be integrated. Moreover, such batch learning prevents scalability as computation time and memory requirements become infeasible for large skill libraries or datasets.[23] Various approaches outside the human–robot interaction scope have addressed these problems.

Initially, the machine learning communities have proposed incremental learning approaches for GMM. Some approaches propose updating a GMM with complete new model component datasets[24] or assume the incoming data points to be time-coherent.[25] Incremental Gaussian Mixture Model learning introduced a way to continuously learn a GMM from an incoming data stream while not fixing the number of total components beforehand.[26,27] Another two-level approach introduces methods for splitting and merging of GMM components.[28] Updating of robotic movement representations online from new demonstrations has also been used for incremental learning of extensions of GMMs for gesture imitation,[23] updating Gaussian Processes from demonstrations and thereby reducing the movement variance[29] or incremental updating of task-parameterized GMM.[30] While all these works focus on updating multiple existing movement representations, in a long-term setting adding new tasks is also important. Approaches that also add new components when needed have been proposed in the context of online updating of task-parameterized semi-tied hidden semi-Markov models for manipulation tasks,[31] learning full-body movements,[32] a bootstrapping cycle for automatic extraction of primitives from complex trajectories[33] or robot table tennis.[34] However, while we draw inspiration from the aforementioned related work, in an HRI scenario it is additionally desirable to consider the inherent coupling and variance in human and robot motions in the demonstrations.

3. Incremental Interaction Primitives

In this section, we present our approach to continuously learn and update multiple cooperative skills from demonstrations. Here, demonstrations are given in form of coupled human and robot trajectories $\boldsymbol{d}_n = \{\boldsymbol{\tau}_n^h, \boldsymbol{\tau}_n^r\}$, where $\boldsymbol{\tau}_n^h$ can, e.g. be a sequence of human wrist positions and $\boldsymbol{\tau}_n^r$ can, e.g. be a sequence of robot joint positions. To learn multiple cooperative tasks from these demonstrations in an online open-ended fashion we introduce a model that is inspired by the Mixture of Experts architecture[35] and consists of two intertwined parts. On the one hand, we use the human trajectories from the demonstrations to train and update a gating model, which will later be used to decide between different cooperative tasks. In addition, we train probabilistic models to generate appropriate robot response trajectories. We deploy Interaction ProMPs,[4] as they are able to capture the inherent correlation in robot and human motions from the demonstrations. Figure 2 summarizes our approach to train this mixture model in an online and open-ended fashion.

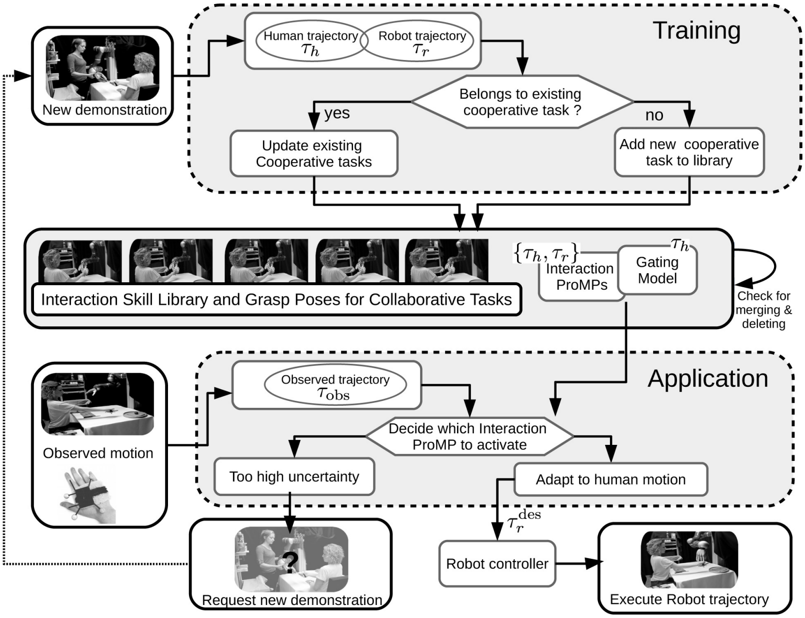


Fig. 2. We introduce a novel approach for online and open-ended learning of a mixture model for cooperative tasks. During training, demonstrations are given in form of trajectories of a human demonstrator $\boldsymbol{\tau}^h$ and corresponding trajectories of a robot arm $\boldsymbol{\tau}^r$, that are obtained via motion capturing and kinesthetic teaching. From these demonstrations, we update or extend the skill library that consists of a gating model and multiple corresponding Interaction ProMPs. During runtime, the gating model decides on activation of particular Interaction ProMPs that we subsequently adapt to the variance in the observed motion. If the gating model is too uncertain about activation of Interaction ProMPs the robot can request more demonstrations.

In the following, we briefly describe the previously proposed batch-based, stationary Mixture of Interaction ProMPs in Sec. 3.1. Next, we present our novel approach to learn a mixture model of Probabilistic Movement Primitives in an online and open-ended fashion in Sec. 3.2. Finally, in Sec. 3.3, we show how the obtained library of multiple interaction ProMPs and the corresponding gating model can be deployed in an HRI scenario.

3.1. *Batch learning for mixture of interaction ProMPs*

Probabilistic Movement Primitives (ProMPs)[17] represent demonstrated movements in the form of distributions over trajectories. In order to obtain this distribution, the trajectories are first approximated by a linear combination of basis functions ϕ. More precisely, a joint position q_t at time step t can be represented as

$$q_t = \boldsymbol{\phi}_t^T \boldsymbol{w} + \epsilon, \tag{1}$$

where $\boldsymbol{\phi}_t$ contains N basis functions ϕ evaluated at time step t, $\boldsymbol{w}$ is a weight vector, ϵ is a zero-mean Gaussian noise and $[\ ]^T$ here, and in the following parts of this paper, denotes transposition of the vector. The choice of basis functions ϕ depends on the type of demonstrated movements. The weight vector $\boldsymbol{w}$ for each demonstrated trajectory is computed with Ridge Regression. For multiple recorded demonstrated trajectories, a Gaussian distribution over the weight vectors $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$ can then be obtained with Maximum Likelihood Estimation. Since the number N of basis functions is usually much lower than the number of time steps of recorded trajectories, the distribution $p(\boldsymbol{w})$ can be seen as a compact representation of the demonstrated movements, which accounts for variability in the execution. In particular, ProMPs offer a representation that allows for operations from probability theory to specify goal or via-points, correlate different degrees of freedom via conditioning and combine different primitives through blending.[17]

An Interaction ProMP[4] is a ProMP that uses a distribution over the trajectories of at least two interacting agents. The demonstrations are now given in the form of a stacked vector for the observed and the controlled agent $\boldsymbol{q} = [\boldsymbol{q}^o, \boldsymbol{q}^c]^T$, where $\boldsymbol{q}^o$ denotes the demonstrated trajectories for the observed agent and $\boldsymbol{q}^c$ denotes the demonstrated trajectories of the controlled agent. Respectively, the weight vector is also represented in an augmented form $\bar{\boldsymbol{w}} = [\boldsymbol{w}_o^T, \boldsymbol{w}_c^T]^T$. Given a set of demonstrations, a distribution over multiple stacked weight vectors can be obtained just as previously described such that $p(\bar{\boldsymbol{w}}) = \mathcal{N}(\mu_{\bar{\boldsymbol{w}}}, \Sigma_{\bar{\boldsymbol{w}}})$. Given a sequence $\mathcal{D}$ of positions of the observed agent (e.g. human), Interaction ProMPs provide methods to infer a corresponding (most likely) trajectory of the controlled agent (robot).[4]

The previously proposed batch learning for Mixture of Interaction ProMPs[5] is an extension to Interaction ProMPs that allows to learn several different interaction patterns from unlabeled demonstrations by applying GMMs, where each mixture component represents one interaction pattern. The Mixture of Interaction Primitives is hereby learned from batch data and the number of components needs to be fixed beforehand. In the case of K different interaction patterns, the distribution over the weight vectors $\bar{\boldsymbol{w}}$ is

$$p(\bar{\boldsymbol{w}}) = \sum_{k=1}^{K} p(k)p(\bar{\boldsymbol{w}}|k) = \sum_{k=1}^{K} \alpha_k \, \mathcal{N}(\bar{\boldsymbol{w}}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2}$$

where α_k is the kth mixture weight that can be prior (if not learned) or posterior (if learned from given data), $\boldsymbol{\mu}_k$ is the mean and $\boldsymbol{\Sigma}_k$ the covariance matrix of the kth component. The parameters of the GMM are hereby learned in the weight space using the EM algorithm. Since this approach assumes that all data is available at the learning time the number of components K remains fixed after learning and the model cannot easily integrate new demonstrations.

3.2. *Online open-ended mixture of interaction ProMPs*

We propose a new method to achieve online learning of cooperative tasks in an open-ended fashion. Hereby, demonstrations are given in the form of robot and human trajectories $\{\boldsymbol{\tau}^r, \boldsymbol{\tau}^h\}$. First, we compute a corresponding representation with weight vectors as introduced in Sec. 3.1. Here, we consider that the human trajectory is of dimensions $D_h \times T$, where D_h is the degree of freedom of the observations (e.g. in case of observing the wrist position $D_h = 3$). T denotes the number of time steps and the robot trajectory is of dimensions $D_r \times T$, where D_r is the degrees of freedom of the robot (e.g., $D_r = 7$ in case of a 7DoF robot arm). For N basis functions ϕ we compute the matrix $\mathbf{\Phi} = [\boldsymbol{\phi}_0, \dots \boldsymbol{\phi}_t, \dots, \boldsymbol{\phi}_T]$ with dimension $N \times T$. In this work, Gaussian basis functions evenly spaced along the time axis are an appropriate choice due to the stroke-based movements. We then compute the weight vectors as a lower-dimensional representation of the trajectories where we first compute the weight vectors for each dimension $\tilde{\boldsymbol{w}}$ as

$$[\tilde{\boldsymbol{w}}_1^h, \dots, \tilde{\boldsymbol{w}}_{D_h}^h, \tilde{\boldsymbol{w}}_1^r, \dots, \tilde{\boldsymbol{w}}_{D_r}^r]^{\mathrm{T}} = (\mathbf{\Phi}\mathbf{\Phi}^T + \beta \boldsymbol{I})^{-1} \mathbf{\Phi} [\boldsymbol{\tau}_h, \boldsymbol{\tau}_r]^{\mathrm{T}}, \tag{3}$$

where β is a factor for Ridge Regression and $\boldsymbol{I}$ is an identity matrix. In experimental evaluation, we found that normalizing the trajectory data within a fixed range before transforming it into the weight space yields overall better result. Subsequently, we compute the stacked weight vectors

$$\boldsymbol{w}_h = [\tilde{\boldsymbol{w}}_1^h, \dots, \tilde{\boldsymbol{w}}_{D_h}^h] \quad \text{and} \quad \boldsymbol{w}_r = [\tilde{\boldsymbol{w}}_1^r, \dots, \tilde{\boldsymbol{w}}_{D_r}^r]. \tag{4}$$

From these demonstrations, now represented in form of $\{\boldsymbol{w}^r, \boldsymbol{w}^h\}$, we learn the two intertwined parts of our model: the gating model that decides on the cooperative tasks based on human motions and multiple corresponding Interaction ProMPs that can subsequently generate a corresponding robot response. For the gating model we train a GMM only on the weights of the human trajectories $\boldsymbol{w}^h$, as at runtime only the human motion will be observed when the system needs to decide on the particular cooperative task and the response of the robot. In parallel to the gating model, the corresponding Interaction ProMPs are trained with the augmented weight vector $\bar{\boldsymbol{w}}$ of human and robot trajectories to model the correlations in the motions.

We assume that new training data needs to be integrated continuously and that we do not know beforehand the number of different collaborative tasks that might be shown to the robot during long-term training. To this end, we use Incremental GMMs[26] to achieve the continuous integration of new demonstrations. Here, we update the gating model and the parameters of the Interaction ProMPs in an EM fashion.

In the Expectation step we compute the responsibilities λ_{kn} of the existing cooperative task k for a new demonstration $\{\boldsymbol{w}_n^h, \boldsymbol{w}_n^r\}$, that is the probability of a new demonstration to belong to an already known cooperative task

$$\lambda_{kn} := p(k|\boldsymbol{w}_n^h) = \frac{p(k)p(\boldsymbol{w}_n^h|k)}{p(\boldsymbol{w}_n^h)} = \frac{\alpha_k \mathcal{N}(\boldsymbol{w}_n^h|\boldsymbol{\mu}_k^g, \mathbf{\Sigma}_k^g)}{\sum_{j=1}^K \alpha_j \mathcal{N}(\boldsymbol{w}_n^h|\boldsymbol{\mu}_j^g, \mathbf{\Sigma}_j^g)}, \tag{5}$$

where $\boldsymbol{\mu}_k^g$ and $\mathbf{\Sigma}_k^g$ are, respectively, the mean and covariance matrix of the kth component of the gating model and α_k are the mixture component weights.

In the Maximization step, we use the responsibilities to recursively update the parameters of the gating model as well as the parameters of the already learned Interaction ProMPs. For each already learned Interaction ProMP k we first compute

$$v_k = v_k + 1, \quad s_k = s_k + \lambda_{kn}, \quad \gamma_k = \frac{\lambda_{kn}}{s_k}, \quad \tilde{\gamma}_k = \gamma_k + \exp(-s_k)\lambda_{kn}, \tag{6}$$

where v_k is the age of the kth component and s_k represents the amount of trajectories the component already modeled well. We then update the parameters of the gating model

$$\begin{aligned}
&\boldsymbol{\mu}_k^g = \boldsymbol{\mu}_k^g + \gamma_k(\boldsymbol{w}_n^h - \boldsymbol{\mu}_k^g), \quad \alpha_k = \frac{s_k}{\sum_{j=1}^K s_j},\\
&\boldsymbol{C}_k^g = (1-\tilde{\gamma}_k)\boldsymbol{C}_k^g + \tilde{\gamma}_k(\boldsymbol{w}_n^h - \boldsymbol{\mu}_k^g)(\boldsymbol{w}_n^h - \boldsymbol{\mu}_k^g)^T - (\tilde{\gamma}_k - \gamma_k)(\boldsymbol{\mu}_k^g - \boldsymbol{\mu}_k^{g,\text{old}})(\boldsymbol{\mu}_k^g - \boldsymbol{\mu}_k^{g,\text{old}})^T,
\end{aligned} \tag{7}$$

where $\boldsymbol{\mu}_k^{g,\text{old}}$ denotes the mean of the gating model before the update. The formulas correspond to the formulas in the incremental GMM,[26] except that we introduce $\tilde{\gamma}_k$ to achieve that during the first demonstrations the covariance is shifted faster away from the (possibly wrong) initialization. Additionally, we compute the updated parameters of the corresponding Interaction ProMPs

$$\begin{aligned}
&\boldsymbol{\mu}_k^e = \boldsymbol{\mu}_k^e + \gamma(\bar{\boldsymbol{w}}_n - \boldsymbol{\mu}_k^e),\\
&\boldsymbol{C}_k^e = (1-\tilde{\gamma}_k)\boldsymbol{C}_k^e + \tilde{\gamma}_k(\bar{\boldsymbol{w}}_n - \boldsymbol{\mu}_k^e)(\bar{\boldsymbol{w}}_n - \boldsymbol{\mu}_k^e)^T - (\tilde{\gamma}_k - \gamma_k)(\boldsymbol{\mu}_k^e - \boldsymbol{\mu}_k^{e,\text{old}})(\boldsymbol{\mu}_k^e - \boldsymbol{\mu}_k^{e,\text{old}})^T,
\end{aligned} \tag{8}$$

where $\boldsymbol{\mu}_k^e$ is the mean of the kth Interaction ProMP and $\boldsymbol{\Sigma}_k^e$ is the covariance matrix of the kth Interaction ProMP. Whenever $p(\boldsymbol{w}_n^h|k)$ is below a threshold T_{nov} for all existing K components we initialize a new component with

$$\begin{aligned}
&\boldsymbol{\mu}_{K+1}^g = \boldsymbol{w}_n^h, \quad \boldsymbol{\Sigma}_{K+1}^g = \boldsymbol{\Sigma}_{\text{init}}^g, \quad \boldsymbol{\mu}_{K+1}^e = \boldsymbol{w}_n, \quad \boldsymbol{\Sigma}_{K+1}^e = \boldsymbol{\Sigma}_{\text{init}}^e, \quad v_{K+1} = 1,\\
&s_{K+1} = 1,
\end{aligned} \tag{9}$$

where $\boldsymbol{\Sigma}_{\text{init}}^g$ and $\boldsymbol{\Sigma}_{\text{init}}^e$ denote the initial covariance matrix of the gating model and the experts that can, e.g. be initialized as identity matrix. If a component has reached a certain age $v_k > v_{\text{min}}^{\text{merge}}$, we check also for merging of components to ensure that no unnecessary components are maintained. Therefore, we compute the probability of the mean of a cluster j to belong to a cluster i as $p(\boldsymbol{\mu}_j|i) = \mathcal{N}(\boldsymbol{\mu}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and decide on merging if $p(\boldsymbol{\mu}_j|i) > T_{\text{merge}}^k$. The threshold for merging is hereby initialized equally with T_{merge} for all new components but increases as a component absorbs more demonstrations, such that already well consolidated components are not so easily merged.

$$T_{\text{merge}}^k = T_{\text{merge}} s_k^{\beta}, \tag{10}$$

where β is a factor that controls how fast component gets consolidated. Once we determined candidates i, j for merging we recompute the joined mean and covariance

$$\boldsymbol{\mu}_{ij} = \frac{s_i\boldsymbol{\mu}_i + s_j\boldsymbol{\mu}_j}{s_i + s_j}, \quad \boldsymbol{\Sigma}_{ij} = \frac{s_i^2\boldsymbol{\Sigma}_i + s_j^2\boldsymbol{\Sigma}_j + (s_i\boldsymbol{\mu}_i + s_j\boldsymbol{\mu}_j)s_i\boldsymbol{\mu}_i + s_j\boldsymbol{\mu}_j^T}{(s_i + s_j)^2} - \boldsymbol{\mu}_{ij}\boldsymbol{\mu}_{ij}^T. \quad (11)$$

We also include a mechanism to delete components created by outliers. Here, we delete all components that do not reach a certain support $s_k > s_{\min}$ after an age $v_k > v_{\min}^{\text{del}}$, such that such outlier components age out over time.

3.3. *A skill library for collaborative tasks*

To demonstrate the use of the learned probabilistic mixture model for cooperative tasks we assume we are now observing the human and obtain an observation $\boldsymbol{w}_*^h$. To determine the most probable cluster given the observations we need to model the posterior of the cluster given the observation $p(k|\boldsymbol{w}_*^h) = \lambda_{k*}$, where λ_{k*} is the responsibility of the kth cluster for the observation $\boldsymbol{w}_*^h$ as defined in Eq. (5). For an observation $\boldsymbol{w}_*^h$ we can now infer the most likely Interaction ProMP k^* using our probabilistic gating model

$$k^* = \arg\max_k \quad p(k|\boldsymbol{w}_*^h). \quad (12)$$

If the responsibility of all components is smaller than the novelty threshold T_{nov} the robot does not execute a response but asks the user for new demonstrations that get subsequently included in the library as described in Sec. 3.2. Otherwise, we condition the chosen component on the observed trajectory to infer the corresponding robot response. Here, the observation $\boldsymbol{o}$ is used to obtain a new Gaussian posterior distribution over the weights, with mean $\boldsymbol{\mu}_{\text{new}}$ and covariance matrix $\boldsymbol{\Sigma}_{\text{new}}$

$$\begin{aligned} \boldsymbol{\Lambda} &= \boldsymbol{\Sigma}_{\boldsymbol{k}^*}\boldsymbol{H}_t(\boldsymbol{\Sigma}_{\boldsymbol{o}} + \boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{k}^*}\boldsymbol{H}_t)^{-1}, \\ \boldsymbol{\mu}_{\text{new}} &= \boldsymbol{\mu}_{\boldsymbol{k}^*} + \boldsymbol{\Lambda}(\boldsymbol{o} - \boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{k}^*}) \qquad \boldsymbol{\Sigma}_{\text{new}} = \boldsymbol{\Sigma}_{\boldsymbol{k}^*} - \boldsymbol{\Lambda}\boldsymbol{H}_t\boldsymbol{\Sigma}_{\boldsymbol{k}^*}, \end{aligned} \quad (13)$$

where $\boldsymbol{\Sigma}_{\boldsymbol{o}} = \boldsymbol{I}\sigma_o$ is the observation noise and $\boldsymbol{H}_t$ is the observation matrix as defined in Ref. 4. More details can be found in Ref. 11. To obtain a corresponding robot motion we execute the mean robot trajectory of this posterior.

4. Experimental Evaluation

We evaluate our approach on 2D trajectory data and on a collaborative scenario with a 7DoF robot arm. For both, we show the qualitative applicability and evaluate the quantitative convergence with respect to a baseline. We demonstrate that the proposed approach can learn personalized libraries for collaborative tasks for different persons and report successful task completion via the decision accuracy of our gating model.

4.1. 2*D trajectory data*

In this section, we demonstrate the application of the proposed incremental learning method on a non-robotic 2D task for letter aquisition. On this 2D data, we visualize the incremental learning process and compare the results to the batch solution of the EM-based method. For the 2D trajectory data experiment, demonstrations are given in the form of multiple hand-drawn letters, as illustrated in Fig. 4(a). All data is normalized to the range of [0, 10] and we apply aequidistant spatial interpolation to decouple the trajectory representation from the varying speed of the demonstrations. Here, we incrementally learn a library of ProMPs. The system never has access to the whole training dataset at once, but only one new unlabeled demonstration is provided at each update step. The general procedure is shown in Fig. 3, where the upper row shows the x-dimension of the learned library and the lower row the accumulated demonstrations. Initially, a single "a" is demonstrated and the first skill is added, with the initial covariance $\boldsymbol{\Sigma}_{\text{init}}$. Additional a's are demonstrated, recognized and used to update the mean and covariance of the corresponding cluster. Once a new demonstration is recognized to not belong to the existing cluster a new cluster is generated. With an increasing number of samples, the variance converges to the variance of the demonstrations as the impact of the initialization covariance decreases. The final skill library consists of five clusters representing the different letters. Please note that in this experiment $\boldsymbol{\mu}_k^g, \boldsymbol{\Sigma}_k^g = \boldsymbol{\mu}_k^e, \boldsymbol{\Sigma}_k^e$. We evaluate the approach in a collaborative setting later in Sec. 4.2.

To demonstrate that the library learned with our new approach using the incremental processing of demonstrations converges to the solution of EM with batch learning, we compare the resulting skill libraries first qualitatively as shown in Fig. 4(c) and quantitatively using the Kullback–Leibler (KL)-Divergence to a baseline as shown in Fig. 4(b). Qualitatively speaking, our approach (Fig. 4(c), upper row) represents all different letters as individual clusters and the trajectory means of

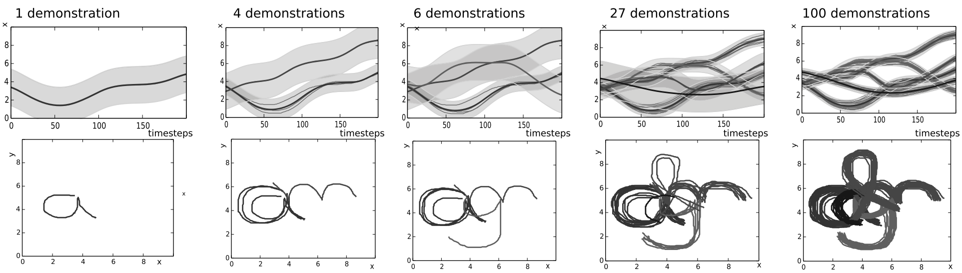


Fig. 3. Learning ProMPs of hand-drawn letter trajectories. All trajectory data is normalized to range of [0, 10]. The demonstrations are normalized to [0, 10] and provided incrementally and no batch data is stored. The intermediate results during training of the ProMP library are shown in the upper row, accumulated demonstrations are shown in the lower row. In the upper row, the shaded area represents two times the standard deviation, the solid lines show the mean, and the demonstrated trajectories are shown as gray lines. Here, our approach successfully updates existing components with new demonstrations and adds new components when required.

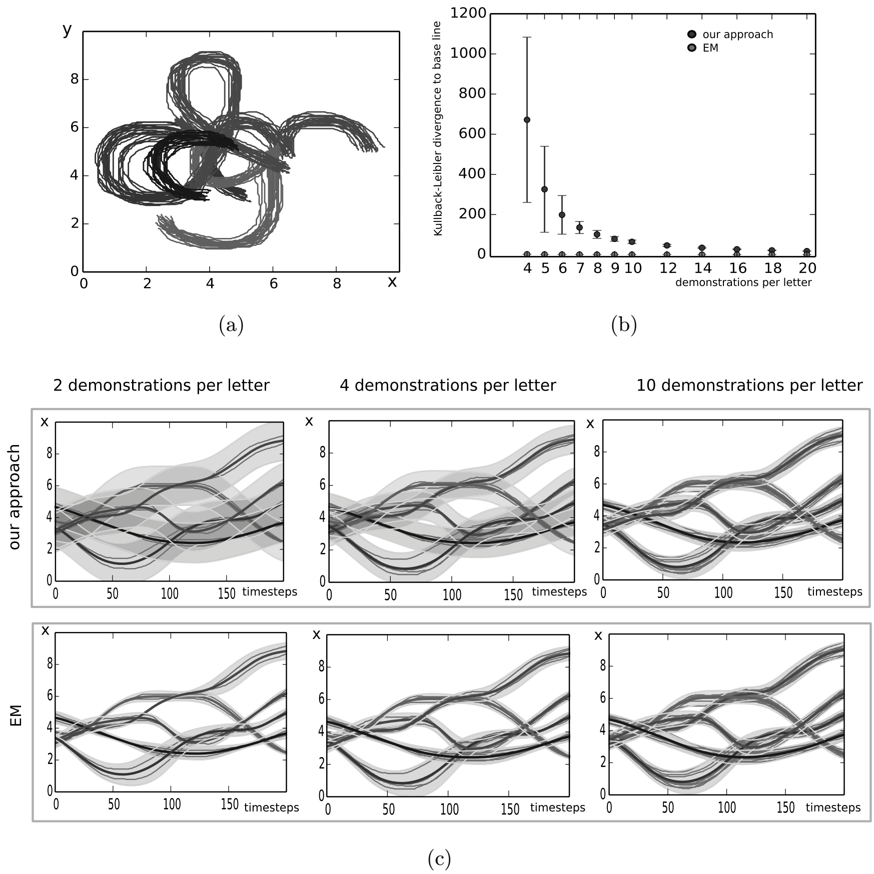


Fig. 4. (a) The demonstrations in the first experiment are given in the form of hand-drawn letter trajectories, normalized to [0, 10]. (b) We compare our approach against an EM approach, where for both we compute the KL-divergence to a baseline solution from labeled data. For increasing number of samples per letter our approach converges against the EM solution, while additionally being able to continuously integrate new data. (c) For increasing number of samples the covariances approximate the underlying data covariances and result in comparable results to the EM approach.

the mixture model components match the means learned with EM in batch mode (Fig. 4(c), bottom row). While for fewer samples per letter the trajectory covariances learned with our approach are dominated by $\mathbf{\Sigma}_{\text{init}}$, with increasing number of samples per letter it approximates the covariances of the EM solution as the influence of the initial covariance decreases. The same behavior can also be observed in the quantitative comparison. Hereby, we compute the (KL)-divergence of our approach and EM to a baseline, computed with Maximum Likelihood estimation from labeled data. The KL-divergence of our approach is averaged over 100 trials, where the order of demonstrations is randomly permuted. In the batch EM case, we provided the method with the correct number of components, while in our approach the algorithm had to find the correct number of components by itself. Figure 4(b) shows that the

KL-divergence between the solution of our approach and the baseline is large for fewer samples and decreases with increasing number of samples. The high variance in the KL-divergence for few samples is expected as the KL-divergence is sensitive to the entropy of the ground truth model, which depends on the selected demonstrations. The variance also shrinks as the entropy converges for multiple samples. The experiments show that our approach achieves results comparable to those of an EM approach. However, the advantage of our approach in contrast to the EM approach is the ability to work on incrementally incoming data while still achieving similar end results. That means our approach does not require all data in batch mode but incrementally learns and updates its models from new demonstrations, which is in particular beneficial in scenarios where not all data is known in the beginning but is only provided over time with new demonstrations.

4.2. *Learning a cooperative tasks with a robotic arm*

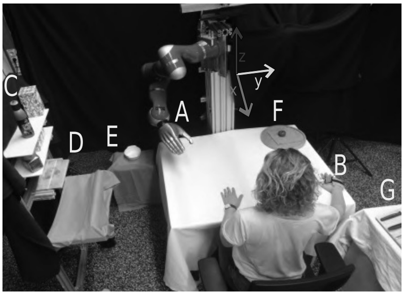
The proposed approach is tested in a collaborative scenario, where a robot is supposed to assist a person in making a salad. The robot assists the person by first observing and recognizing the human action and second determining, adapting and executing an assistive response based on prior demonstrations. For the salad scenario, shown in Fig. 5(a), five different cooperative tasks are required, namely:

- Board: robot hands over the cutting board after human grasped the knife;
- Tomato: robot passes the tomato when human reaches for the tomato;
- Bowl: robot passes the salad bowl when human reaches for the bowl;
- Dressing: robot gets the salad dressing from the shelf for the human;
- Standup: robot supports the standup motion of the human.

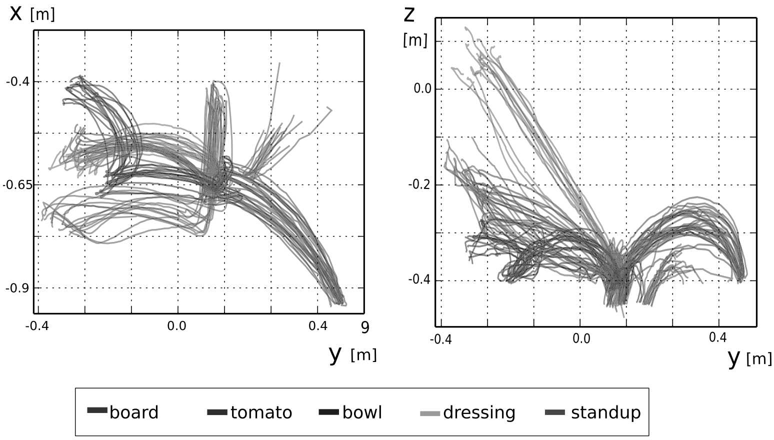
Each of the cooperative tasks is demonstrated separately and multiple times. To record the demonstrations for the collaborative task we use sensory informations of a motion capturing system and the joint encoders of the robot. Thereby, the robot trajectory is shown by kinesthetic teaching, while the human action is recorded using motion capturing markers attached to the wrist (pink bracelet Fig. 5(c)). The used motion capturing system provides position and orientation of the human wrist (streamed to a ROS node) with an accuracy up to 1 cm at a framerate of 150 Hz. For the experiments in this paper only the position was used. Additionally, for each skill a fixed grasp pose for the corresponding object is added to the robots skill database. In general, it would be beneficial to include also haptic feedback for grasping and camera based object tracking, however since this is not the main focus of this paper only fixed object positions and grasp positions where used in an open loop fashion. The teaching is shown for the bowl task in Fig. 6 (top row).

4.2.1. *Learning personalized skill libraries*

In an initial experiment, 15 demonstrations are recorded for every task with a test subject (subject 1). The resulting human trajectories are shown in top–down and



(a)



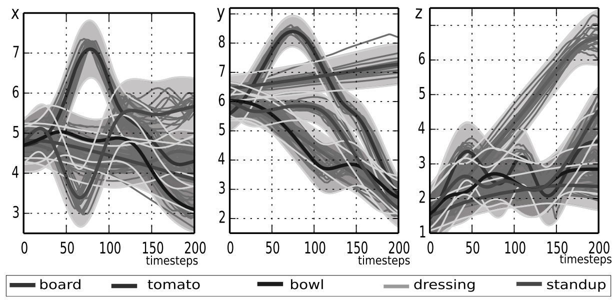
(b)

(c)

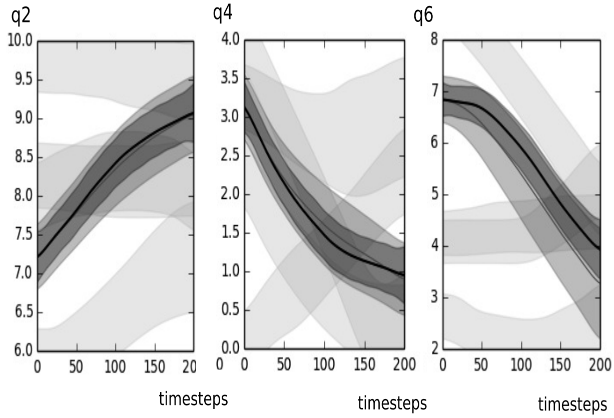
Fig. 5. (a) We evaluate our approach in a collaborative task where a robot (A) assists a human (B) in making a salad. The robot can hand over the board (D) the dressing (C), a tomato (F), the bowl (E) or assist with a standup motion. (b) For a human test subject (subject 1) we recorded 15 demonstrations per task. The trajectories are shown from top–down and front view. (c) We incorporate an automatic calibration procedure to determine the individual workspace boundaries of different subjects when subjects perform a circle in x–y plane (blue) and in y–z plane (green).

Fig. 6. (upper row) Demonstrations are recorded as human and robot trajectories. For the robot we use kinesthetic teaching and the human wrist trajectory is tracked with a motion capturing markers. (lower row) The robot performs motions according to recognized previous demonstrations.

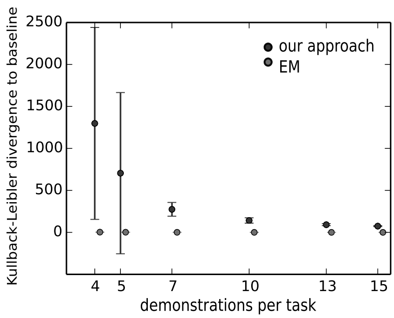
front view in Fig. 5(b). The recorded data is normalized to the range of [0, 10] and we apply aequidistant spatial interpolation to decouple the trajectory representation from the varying speed of the demonstrations. From the demonstrations our approach learns a cooperative skill library consisting of a gating model and multiple corresponding Interaction ProMPs. Hereby, the demonstrations are not provided as batch data but incrementally. An example of the gating model (which corresponds to the human part of the Interaction ProMPs) is shown in Fig. 7(a). Five different skill clusters are clearly visible. Figure 7(c) shows that similarly to the letter experiment, the averaged KL-divergence with respect to the ground truth solution learned from labeled data decreases with the number of demonstrations per task and is for more demonstrations comparable to the EM solution, while it enables incremental updates on new incoming data and does not require to store or recompute all batch data. In the interactive setting, the robot adapts its response to the human movement based on prior demonstrations. The output of the skill library is hereby a desired joint trajectory of the robot that gets executed by the robot's trajectory controller. Such an adapted robot response is shown for the bowl task in Fig. 6 in the bottom. The adaptation of the robot response is achieved by conditioning the Interaction ProMP on the human trajectory as described in Sec. 3.3. An example of such an adaptation can be seen in Fig. 7(b), for the tomato task. To further evaluate the applicability and robustness of the proposed approach, we conducted more experiments with different subjects and identical hyperparameters. For each subject individual demonstrations are recorded and a corresponding personalized skill library is incrementally learned. To evaluate the performance, the classification accuracy for recognizing the correct cooperative task is evaluated by k-fold cross-validation. For Subject 1 we use a training set of 10 demonstrations per task and test on 5 demonstrations per task, while all other subjects use 4 demonstrations per task for training and 1 demonstration per task for testing. The classification results averaged over 100 test and train sets are shown in Table 1. The first value corresponds to the percentage of successful classifications, the second to the percentage of wrong



(a) Gating model for human trajectories (wrist trajectory distribuions)



(b) Robot trajector distributions (robot joints 2, 4 and 6)



(c)

Fig. 7. (a) The gating model, which corresponds to the human trajectory part of the Interaction ProMPs (data normalized to [0, 10], shaded area is two times the standard deviation, the solid lines show the mean, gray lines show demonstrated trajectories).(b) Our model produces a corresponding robot response to an observed human trajectory. The gating model decides which Interaction Primitive (light gray) to activate (green). The activated primitive is adapted to the variance in the observed human trajectory via conditioning (dark gray). The plots show joints q2, q4 and q6 of the robot (shaded area represents two times standard deviation, the solid lines show the mean) (c) We compare the KL-divergence of our and an EM approach to a baseline from labeled data.

Table 1. Classification accuracy.

	Board	Tomato	Dressing	Standup	Bowl
Subject1	1.0	1.0	1.0	1.0	0.99 (0|0.01)
Subject2	1.0	1.0	1.0	1.0	1.0
Subject3	0.85 (0|0.15)	1.0	1.0	0.72 (0|0.28))	1.0
Subject4	1.0	1.0	1.0	1.0	1.0
Subject5	1.0	1.0	0.99 (0.01|0)	1.0	1.0
Subject6	1.0	1.0	1.0	1.0	1.0
Subject7	0.77 (0|0.23)	1.0	0.88 (0|0.12)	1.0	1.0
Subject8	1.0	1.0	1.0	1.0	1.0
Subject9	0.84 (0|0.16)	1.0	1.0	1.0	1.0
Subject10	1.0	1.0	1.0	0.77 (0.23|0)	1.0

Algorithm 1. Incremental Skill Learning

input: $\boldsymbol{\Sigma}^g_{\text{init}} = \boldsymbol{I}, T_{\text{nov}}, v_{\min}$
while new data $\boldsymbol{\tau}^h_n$,$\boldsymbol{\tau}^r_n$ **do**
    normalize data , Eq. (14)
    compute $\boldsymbol{w}^h_n$,$\boldsymbol{w}^r_n$ from $\boldsymbol{\tau}^h_n$,$\boldsymbol{\tau}^r_n$ Eqs. (3) and (4)
    compute $p(\boldsymbol{w}^h_n|k)\ \forall k$
    if $p(\boldsymbol{w}^h_n|k) < T_{\text{nov}}\ \forall k$ **then**
        add new component Eq. (9); k++
    else
        compute $p(k|\boldsymbol{w}^h_n)\ \forall k$, Eq. (5).
        update $\forall k$ Eqs. (6), (7) and (8)
        if $v_k > v_{\min}$ **then**
            check for merge, Eqs. (10) and (11)

classifications (e.g. tomato as bowl) and the third to the percentage of classifications as unknown. The results reveal that our approach works well for six of the subjects but classification accuracies for the other 4 subjects vary between tasks. For the board task Subject 3, 7 and 9 have some movements with a high variance to the training set that are classified as unknown. However, the classification as unknown does not yield a wrong robot response and the robot would only ask for a new demonstration. Only for Subject 10 the robot misclassifies the standup skill. Table 2 shows the number of learned components for the individual subjects. Depending on a variance of a subject's movement a single skill can be represented by multiple clusters since we used the same hyperparameters for all subjects and did not tune them individually. Additional clusters do not cause wrong classifications but can lead to unknown classification (Subject 3, 7 and 9). Results for Subject 10 show that the wrong classifications for standup were due to too few learned clusters.

Table 2. Clusters per subject.

	4	5	6	7
Sub1	0	1.0	0	0
Sub2	0	1.0	0	0
Sub3	0	0.23	0.77	0
Sub4	0	1.0	0	0
Sub5	0.01	0.72	0.23	0.04
Sub6	0	1.0	0	0
Sub7	0	0.85	0.15	0
Sub8	0	1.0	0	0
Sub9	0	0.78	0.22	0
Sub10	0.23	0.77	0	0

4.2.2. *Data pre-processing for skill transfer between subjects*

We apply two steps of data pre-processing to ensure better transfer of learned skills between different subjects. First, we normalize the demonstrated trajectories with respect to the personal workspace boundaries of the subjects. These boundaries are extracted out of calibration data, where we let the subjects perform half circular movements in the x–y and z–y plane as depicted in Fig. 5(c) and record their wrist positions. From the recorded data we extract maximum and minimum values for each task space dimension $\{\boldsymbol{X}_{\min}, \boldsymbol{X}_{\max}\}$. For the z dimension, we noticed using only 90% of the extracted boundaries works well in our setup since subjects tend to stretch out more during calibration than when executing actual gestures. Given the personal boundaries, we normalize demonstrated trajectories to the range [0, 10] and subtract the normalized start position of the trajectory

$$\boldsymbol{\tau}^n_{\text{norm}} = \frac{10(\boldsymbol{\tau}^n - \boldsymbol{X}_{\min})}{\boldsymbol{X}_{\max} - \boldsymbol{X}_{\min}} - \boldsymbol{\tau}^0_{\text{norm}}, \tag{14}$$

where $\boldsymbol{\tau}^n_{\text{norm}}$ is the normalized nth point of the trajectory and $\boldsymbol{\tau}^0_{\text{norm}}$ is the first normalized point of the trajectory. Here, we assume that the overall position of the human is static (e.g. seated on a chair), but exact start positions of the hand during task execution might differ in between trial or subject.

To evaluate the data pre-processing we use four trajectories per task per subject and incrementally compute a skill library per task. We randomize the ordering of trajectories and average the number of resulting components per task over 100 random seeds. The results are shown in Table 3, and Fig. 8 illustrates exemplary

Table 3. Number of clusters per task across subjects, with and (without) transformation.

	1	2	3	4	5	≥ 6
Tomato	1.0 (0.84)	0. (0.11)	0. (0.04)	0. (0.01)	0.(0.)	0.(0.)
Board	0. (0.)	0.02(0.)	0.27 (0.0)	0.42 (0.0)	0.27 (0.12)	0.02 (0.88)
Dressing	0.43 (0.)	0.39 (0.)	0.12 (0.03)	0.03 (0.11)	0.03 (0.3)	0. (0.56)
Bowl	0.06 (0.)	0.75 (0.0)	0.18 (0.)	0.01 (0.07)	0. (0.3)	0. (0.63)
Standup	0. (0.)	0. (0.)	0. (0.)	0. (0.)	0.06 (0.)	0.94 (1.)

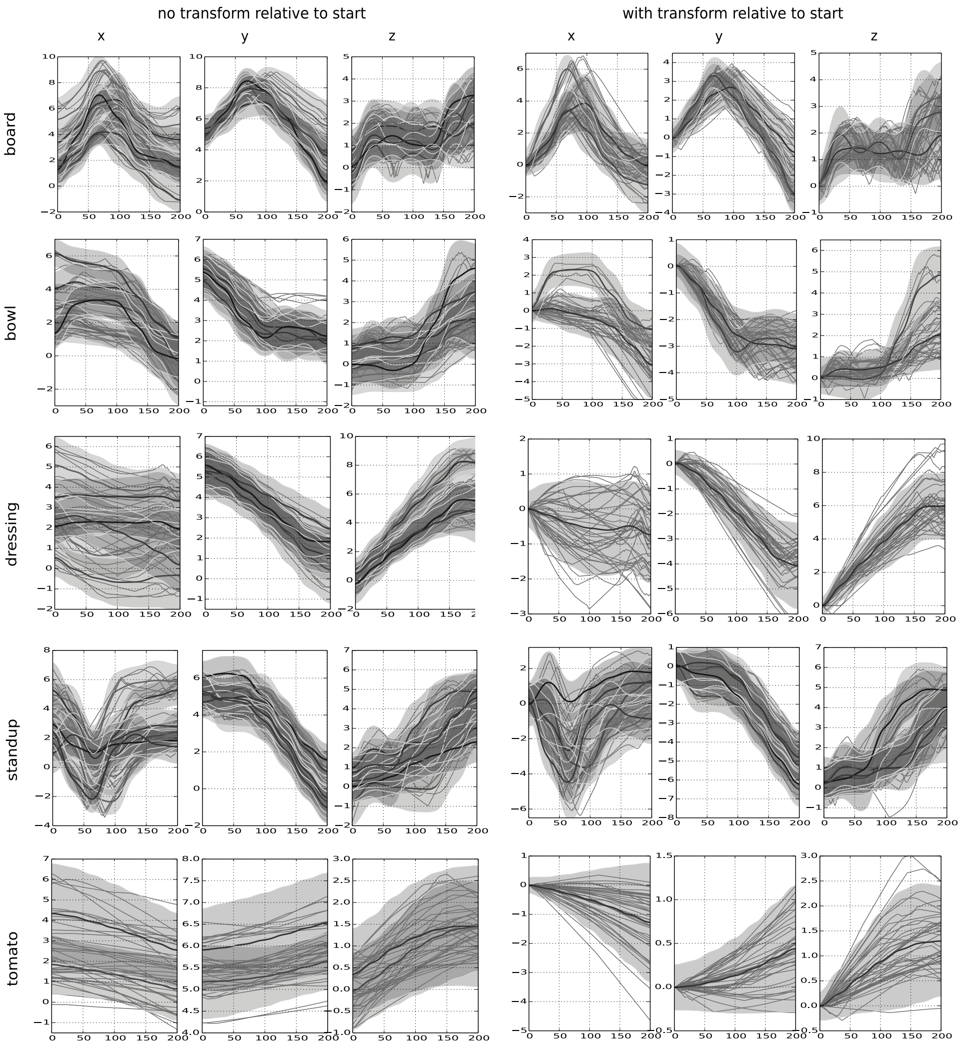


Fig. 8. We evaluate how the human motions (normalized data) for the different tasks vary in between subjects and how many components a skill library trained in incremental fashion learns for each task across subjects. The transformation of human trajectories relative to the start position results in higher similarity of trajectories across subjects and therefore less components per task.

results of the resulting skill library components. In particular, we compare results without transformation relative to the start point and results with transformation. The experiments show that in general the transformation results in better generalization in between subjects and therefore less components per task. While classification still works for too many components this can result in problems when trying to apply a library to unseen subject data. It also shows that while the tasks dressing, tomato and bowl generalize well and mainly result in only one or two components

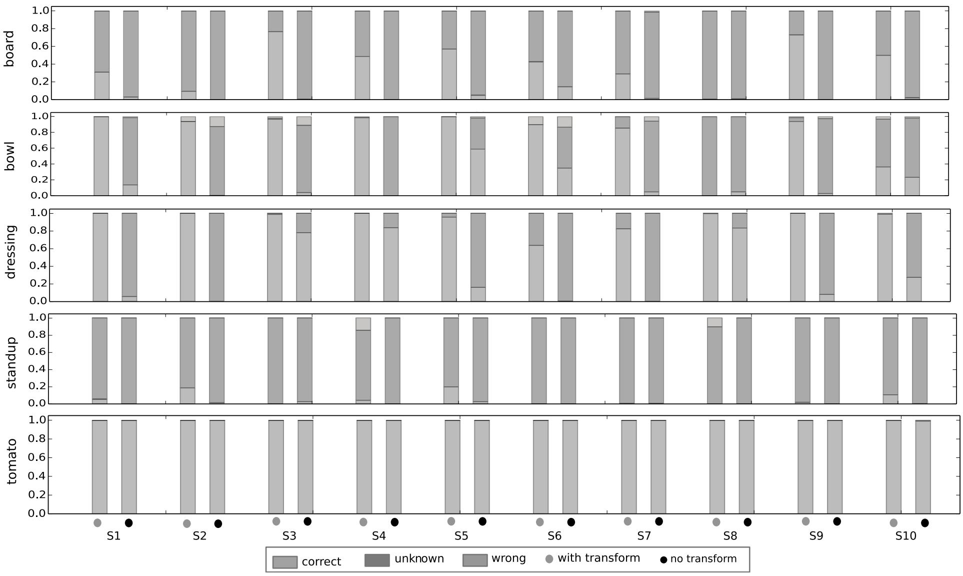


Fig. 9. We test a skill library learned on nine of ten subjects on the unseen tenth subject. It shows that the transformation of the data yields better results in terms of percentage of correct classifications (green), unknown classifications (blue) and wrong classifications (red) averaged over 100 combinations of test and training sets. It also shows that not all tasks generalize equally well.

over all subjects, the task board and in particular standup result in multiple different model components and do not generalize so well for different subjects.

4.2.3. *Evaluation of the skill libraries for unseen subjects*

In this section, we evaluate how a library trained on data across 9 subjects performs in classification of trajectories from a new unseen subject. We train hereby on 5 trajectories per task per subject and test for 5 trajectories per task on the unseen subject. Hereby, we randomize the order of the training trajectories and average over 100 trials per unseen subject. Again we compare results with and without transformation of the trajectory data. Figure 9 shows the resulting accuracies for each subject. It shows that the transformation in general leads to better classification accuracy. While the transfer of the learned library to unseen subjects works well for tomato, dressing and bowl task for most of the subjects, the board and standup task do not transfer so well. Eventhough there are only little wrong classifications for standup and board there is a high percentage of unknown classifications which shows the trajectories of the new subject are too different from the learned library. Our approach could handle this by creating new model components for such cases. The results indicate that shorter motions, such as tomato, bowl or dressing, generalize better in between subjects. For more complex motions it might be beneficial to consider a different trajectory representation or additional modalities in the future.

4.3. *Limitations*

So far we use a gating model based on wrist motions of the human and their geometric representation. For better distinguishability of more complex gestures it would be important to also consider full arm motions and dynamics, for which the gating could be exchanged by another representation. We think this missing complexity of the gating so far limits scalability of our method to a large number of different tasks and could be tackled in future work. Additionally, our experimental setup currently relies on the high accuracy of a motion capturing sytem. Another line of work is therefore to include camera based tracking of the human and objects, and compare the results of our method regarding, e.g. reduced tracking accuraccy. Including also haptic sensory feedback and more advanced representations for grasping would be also beneficial for the skill library from a practical point of view.

5. Conclusions

In this paper, we introduce a novel approach to learn a mixture model of probabilistic interaction primitives in an online and open-ended fashion. In contrast to existing batch approaches our approach is able to update existing interaction primitives continuously from new data and extend a cooperative tasks library with new interaction patterns when needed. Experimental evaluation on a collaborative scenario with a 7DoF robot arm showed that our approach is able to learn multiple different collaborative tasks from unlabeled training data and generate corresponding robot motions, based on prior demonstrations. Additionally, evaluations with 10 human subjects showed that our approach successfully learned a personalized collaborative library for the majority of subjects. Moreover, we evaluated how a library trained across multiple subjects generalizes to unseen new subjects.

However, since the experiments on different subjects indicate that motion data do not work equally well for all subjects and tasks, we are currently investigating how to include other modalities such as gaze direction or voice commands in our gating model. Another line of research is to online adapt the hyperparameters, which are currently only hand-tuned, automatically to individual subjects and investigate more principled ways of hyperparameter selection, which can potentially improve classification results for individual subjects. Moreover, since for now the Interaction ProMPs in the cooperative tasks library are solely learned from demonstrations an important component for future work is to enrich and improve the trajectories of the robot, for example, by using reinforcement learning and include more sensory channels such as camera-based perception instead of the motion capturing system, and additional haptic sensing for more advanced interaction skills.

Acknowledgments

This work is funded by the German Federal Ministry of Education and Research (BMBF) project 16SV7984 (KOBO34), by ERC StG 640554 (SKILLS4ROBOTS).

References

1. K. Linz and S. Stula, Demographic change in Europe-an overview, *Observatory for Sociopolitical Developements in Europe* **4**(1) (2010) 2–10.
2. S. Schaal, Is imitation learning the route to humanoid robots? *Trends Cogn. Sci.* **3**(6) (1999) 233–242.
3. D. A. Rosenbaum, *Human Motor Control* (Academic press, 2009).
4. G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters and G. Neumann, Learning interaction for collaborative tasks with probabilistic movement primitives, in *2014 14th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)* (IEEE, 2014), pp. 527–534.
5. 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 *2015 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2015), pp. 1535–1542.
6. D. Koert, S. Trick, M. Ewerton, M. Lutter and J. Peters, Online learning of an open-ended skill library for collaborative tasks, in *2018 IEEE-RAS 18th Int. Conf. Humanoid Robots (Humanoids)* (IEEE, 2018), pp. 1–9.
7. B. D. Argall, S. Chernova, M. Veloso and B. Browning, A survey of robot learning from demonstration, *Robot. Auton. Syst.* **57**(5) (2009) 469–483.
8. A. Billard, S. Calinon, R. Dillmann and S. Schaal, Robot programming by demonstration, in *Springer Handbook of Robotics* (Springer, 2008), pp. 1371–1394.
9. P. Pastor, H. Hoffmann, T. Asfour and S. Schaal, Learning and generalization of motor skills by learning from demonstration, in *2009 IEEE Int. Conf. on Robotics and Automation* (IEEE, 2009), pp. 763–768.
10. 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 *2017 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2017), pp. 2882–2889.
11. G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer and J. Peters, Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks, *Auton. Robots* **41**(3) (2017) 593–612.
12. H. B. Amor, D. Vogt, M. Ewerton, E. Berger, B. Jung and J. Peters, Learning responsive robot behavior by imitation, in *2013 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IEEE, 2013), pp. 3257–3264.
13. D. Vogt, B. Lorenz, S. Grehl and B. Jung, Behavior generation for interactive virtual humans using context-dependent interaction meshes and automated constraint extraction, *Comput. Animation Virtual Worlds* **26**(3–4) (2015) 227–235.
14. S. Nikolaidis, R. Ramakrishnan, K. Gu and J. Shah, Efficient model learning from joint-action demonstrations for human-robot collaborative tasks, in *Proc. 10th Annual ACM/IEEE International Conf. Human-Robot Interaction* (ACM, 2015), pp. 189–196.
15. A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor and S. Schaal, Dynamical movement primitives: Learning attractor models for motor behaviors, *Neural Comput.* **25**(2) (2013) 328–373.
16. S. Calinon, F. Guenter and A. Billard, On learning, representing, and generalizing a task in a humanoid robot, *IEEE Trans. Syst. Man, Cybernet. Part B (Cybernetics)* **37**(2) (2007) 286–298.
17. A. Paraschos, C. Daniel, J. Peters and G. Neumann, Using probabilistic movement primitives in robotics, *Auton. Robot.* **42**(3) (2018) 529–551.
18. Y. Huang, L. Rozo, J. Silvério and D. G. Caldwell, Kernelized movement primitives, *The Int. J. Robot. Res.* **38**(7) (2019) 833–852.
19. H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer and J. Peters, Interaction primitives for human-robot cooperation tasks, in *2014 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2014).

20. C. Pérez-D'Arpino and J. A. Shah, Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification, in *2015 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, 2015), pp. 6175–6182.
21. D. Lee, C. Ott and Y. Nakamura, Mimetic communication model with compliant physical contact in human-humanoid interaction, *The Int. J. Robot. Res.* **29**(13) (2010) 1684–1704.
22. G. Konidaris, S. Kuindersma, R. Grupen and A. Barto, Robot learning from demonstration by constructing skill trees, *The Int. J. Robot. Res.* **31**(3) (2012) 360–375.
23. S. Calinon and A. Billard, Incremental learning of gestures by imitation in a humanoid robot, in *Proc. ACM/IEEE Int. Conf. Human-Robot Interaction* (ACM, 2007), pp. 255–262.
24. A. Ahmed and E. Xing, Dynamic non-parametric mixture models and the recurrent chinese restaurant process: With applications to evolutionary clustering, in *Proc. 2008 SIAM Int. Conf. Data Mining* (SIAM, 2008), pp. 219–230.
25. O. Arandjelovic and R. Cipolla, Incremental learning of temporally-coherent gaussian mixture models, *Society of Manufacturing Engineers (SME) Technical Papers* (British Machine Vision Conference, 2005), pp. 1–1.
26. P. M. Engel and M. R. Heinen, Incremental learning of multivariate gaussian mixture models, in *Brazilian Symp. Artificial Intelligence* (Springer, 2010), pp. 82–91.
27. R. C. Pinto and P. M. Engel, A fast incremental gaussian mixture model, *PloS one* **10**(10) (2015) e0139931.
28. A. Declercq and J. H. Piater, Online learning of gaussian mixture models-a two-level approach. in *VISAPP (1)*, 2008, pp. 605–611.
29. G. Maeda, M. Ewerton, T. Osa, B. Busch and J. Peters, Active incremental learning of robot movement primitives, in *Conf. Robot Learning (CORL)*, 2017.
30. J. Hoyos, F. Prieto, G. Alenya and C. Torras, Incremental learning of skills in a task-parameterized gaussian mixture model, *J. Intell. Robot. Syst.* **82**(1) (2016) 81–99.
31. I. Havoutis, A. K. Tanwani and S. Calinon, Online incremental learning of manipulation tasks for semi-autonomous teleoperation, in *IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), Workshop on Closed-loop Grasping and Manipulation* (Challenges and Progress, Daejeon, Kora, 2016).
32. D. Kulić, C. Ott, D. Lee, J. Ishikawa and Y. Nakamura, Incremental learning of full body motion primitives and their sequencing through human motion observation, *The Int. J. Robot. Res.* **31**(3) (2012) 330–345.
33. A. Lemme, R. F. Reinhart and J. J. Steil, "Self-supervised bootstrapping of a movement primitive library from complex trajectories," in *2014 14th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)* (IEEE, 2014), pp. 726–732.
34. K. Muelling, J. Kober and J. Peters, Learning table tennis with a mixture of motor primitives," in *2010 10th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)* (IEEE, 2010), pp. 411–416.
35. R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive mixtures of local experts," *Neural comput.* **3**(1) (1991) 79–87.

Dorothea Koert received her M.Sc. degrees in Computational Engineering from the Technical University of Darmstadt, Germany, in 2016. She is a Ph.D. student at the Institute of Intelligent Autonomous Systems, and works on an assistive robot for elderly persons. Her research includes Learning from Demonstrations, Human–Robot Interaction, and Mixture Models.

Susanne Trick received her M.Sc. degree in Psychology in IT from the Technical University of Darmstadt, Germany in 2018. She is a Ph.D. student at the department Psychology of Information Processing at TU Darmstadt working on Computational Modelling of human behavior, Human-Robot Interaction, and Intention Recognition.

Marco Ewerton received his M.Sc. degree in Electrical and Information Engineering from the Technical University of Darmstadt, Germany in 2014. He is a Ph.D. student at the department of Intelligent Autonomous systems at TU Darmstadt. His research interests include Imitation Learning, Human-Robot Interaction, and Motor Skill Learning.

Michael Lutter received his M.Sc. degree in Masters in Electrical Engineering from the Technical University of Munich, Germany, in 2016. He is a Ph.D. student at the department of Intelligent Autonomous systems at TU Darmstadt working on Autonomous Robotics and Reinforcement Learning.

Jan Peters is a full professor (W3) for Intelligent Autonomous Systems at the Computer Science department of Technical University Darmstadt and adjunct senior research scientist at the Max-Planck Institute for Intelligent Systems. He graduated from University of Southern California with a Ph.D. in Computer Science in March 2007. His research interests include robotics, nonlinear control, machine learning, and motor skill learning.