



# Enhancing learning capabilities of movement primitives under distributed probabilistic framework for flexible assembly tasks

Likun Wang<sup>1</sup> · Shuya Jia<sup>2</sup> · Guoyan Wang<sup>3</sup> · Alison Turner<sup>1</sup> · Svetan Ratchev<sup>1</sup>

Received: 3 January 2021 / Accepted: 14 September 2021 / Published online: 1 October 2021  
© The Author(s) 2021

## Abstract

This paper presents a novel probabilistic distributed framework based on movement primitives for flexible robot assembly. Since the modern advanced industrial cell usually deals with various scenarios that are not fixed via-point trajectories but highly reconfigurable tasks, the industrial robots used in these applications must be capable of adapting and learning new in-demand skills without programming experts. Therefore, we propose a probabilistic framework that could accommodate various learning abilities trained with different movement-primitive datasets, separately. Derived from the Bayesian Committee Machine, this framework could infer new adapting trajectories with weighted contributions of each training dataset. To verify the feasibility of our proposed imitation learning framework, the simulation comparison with the state-of-the-art movement learning framework task-parametrised GMM is conducted. Several key aspects, such as generalisation capability, learning accuracy and computation expense, are discussed and compared. Moreover, two real-world experiments, i.e. riveting picking and nutplate picking, are further tested with the YuMi collaborative robot to verify the application feasibility in industrial assembly manufacturing.

**Keywords** Learning from demonstration · Task-parametrised · Probabilistic distributed framework · Bayesian committee machine · Assembly

## 1 Introduction

In modern advanced manufacturing, the industrial robots are widely used in assembly tasks, such as peg-in-hole [7, 27], slide-in-the-groove [18], bolt screwing [11, 12] and pick-and-place [10, 24]. Owing to high-precision sensors, leading driven techniques and excellent mechanical structure, industrial robots can successfully deal with known objects within the well-structured assembly environment. However, current industrial robots can hardly handle complex assembly processes or adapt to unexpected changes. To maintain a robust control performance,

industrial robots are usually programmed to follow fixed trajectories, especially for large workpiece assembly. For flexible manufacturing applications, industrial robots are required to perform several tasks with various end-effectors regarding different assembly environments. Therefore, assistant measurement devices, i.e. machine vision system and metrology [15], could provide a reference target for the robots. Nevertheless, they can only be applied in a certain region of interest, which more or less limits the generalisation of retrieving novel trajectories.

Generally, the core idea of assembly is to generate ordered operations consisting of a set of movement primitives, which could bring individual components together to produce a novel product. Similarly, an excellent operator does have the prime skills in terms of performing assembly tasks, which promotes a feasible scenario for robots to learn from human demonstration.

In the context of learning from demonstration, several algorithms, i.e. probabilistic movement primitives (ProMP) [17] and dynamic movement primitives (DMP) [20], have been proposed to generate desired trajectories regarding

---

✉ Likun Wang  
Likun.Wang@nottingham.ac.uk

<sup>1</sup> Centre for Aerospace Manufacturing, University of Nottingham, Nottingham, United Kingdom

<sup>2</sup> School of Computer Science and System, Bauman Moscow State Technical University, Moscow, Russia

<sup>3</sup> Manufacturing Department, Safran Landing System, Gloucester, UK

different modulations. Both ProMP and DMP introduce various weight coefficients to describe basis functions and govern explicit dynamic equations, separately. As a time-driven algorithm, the weight parameters of the basis function are learned towards an optimal function value without addressing high-dimensional inputs.

In order to address high-dimensional issues and alleviate specified trajectory equations, Gaussian Mixture Model (GMM) [3] is applied to model several Gaussian distributions of demonstrations probabilistically using the EM algorithm. Combining with Gaussian Mixture Regression (GMR) [4], the novel predicted trajectories are derived from a weighted conditional Gaussian distribution. However, the capability of generating trajectories is limited by the similarity (Euclidean distance in the covariance function) [26] of the demonstration and the desired input. A similar kernel-based framework, such as movement primitives with multi-output Gaussian Process [6] and Kernelised Movement Primitives [5], could be seen as the variations of GMM/GMR, which take advantage of the kernel function to retrieve more flexible trajectories.

Reinforcement learning is considered as an alternative to adapt new tasks according to the optimisation reward. In [22], Policy Improvement with Path Integrals (PI<sup>2</sup>) is used to refine the movement primitives of DMP. A modified version of PI<sup>2</sup> based on Monte Carlo Sampling is introduced in [21] to enhance the learning performance. Additionally, Q learning algorithm, such as nature actor-critic [19], is applied for automatically selecting the centres of GMM clusters. Nevertheless, the learning procedure based on sampling optimisation might be time-consuming.

Although robots are usually supposed to generate feasible trajectories in a wide range of various circumstances, human demonstrations could only provide limited sets of learning instances. Therefore, in addition to the above-mentioned imitation learning algorithms, several modified versions have been proposed to add more advanced properties in order to enhance the capability of generating adapting trajectories. In [14], based on ProMP, a probabilistic human–robot interaction methodology is proposed in collaboration with an operator. Moreover, the spring-damper dynamic behaviour regarding impedance control is discussed in [9]. A task-parametrised formulation extended from GMM is presented in [2], which essentially models movement behaviours with a set of task parameters, and therefore improving generalisation capability.

The remainder of the paper is organised as follows: after the introduction, an overview of the distributed probabilistic framework is presented in Sect. 2; Additionally, Sect. 3 outlines the individual movement primitive learning of GMM clustering and GMR regression with the EM algorithm; in Sect. 4, the multiple movement primitives

learning under the distributed regression framework is addressed; Sect. 5 presents the comparison between the task-parametrised GMM and our proposed learning framework, along with several assembly tasks using ABB YuMi robot in order to verify the application feasibility; finally, the conclusion is reported in Sect. 6.

## 2 Distributed probabilistic framework—an overview

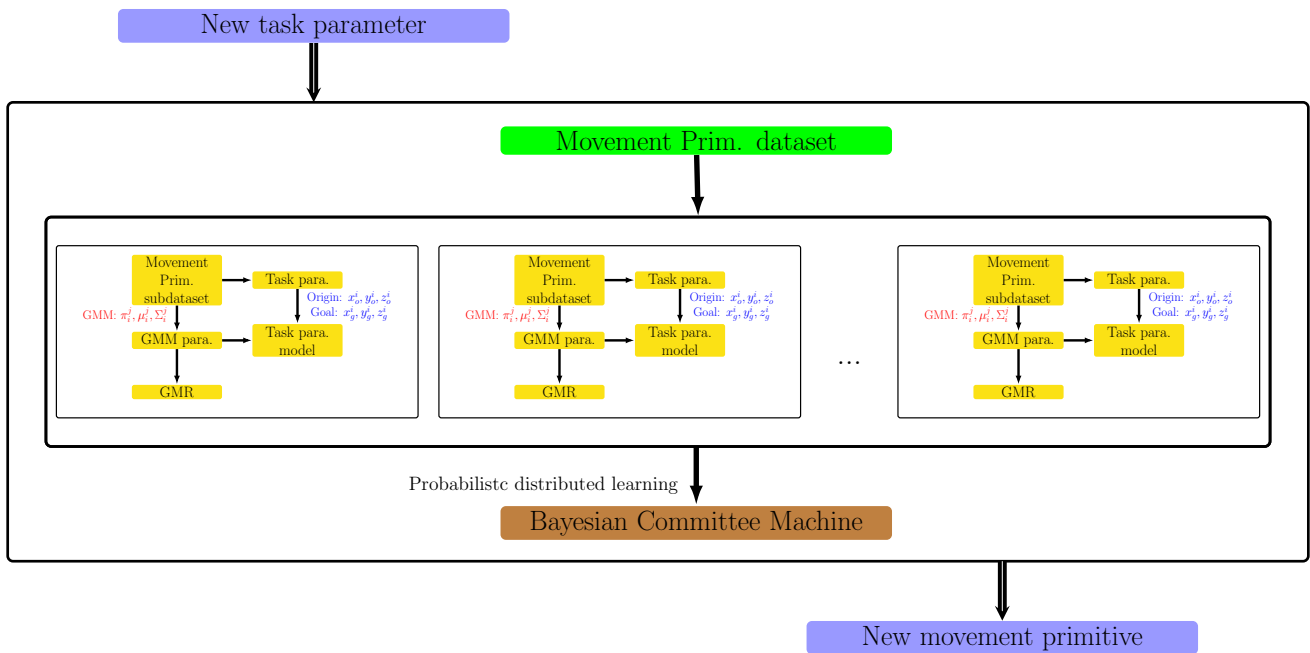
Nearly all the movement-primitive imitation learning methods focus on the adaptation and modulation of a single human demonstration template. As usual, these human demonstrations are captured under specific conditions such as obstacle constraints, limited sensor devices or with a redundant manipulator. All these facts would place barriers in the way of reconfiguring or retrieving novel trajectories regarding a different task setting.

Therefore, in this paper, we propose a novel distributed probabilistic framework for enhancing the learning capabilities among different movement primitives. More specifically, as illustrated in Fig. 1, this framework aims to accommodate different movement primitives by storing the task parameters, along with primitives parameters obtained by GMM and GMR. Both parameters are further utilised to establish a nonlinear mapping based on the Gaussian process.

Furthermore, the Bayesian Committee Machine is employed as a probabilistic fusion machine to automatically choose a training movement primitive of retrieving a new movement primitive given the combination of several training movement primitives (adaptation). The core idea of our proposed framework is that it preserves the individual functions and features of each movement primitive, and meanwhile flexibly outputs novel motions that meet the demand of the task environment.

To improve the readability of this paper, we highlight our contributions as follows:

1. We propose a novel distributed probabilistic framework, which could accommodate various movement-primitive datasets into an overall regression structure.
2. Based on the Evidence Maximisation, the hyper-parameters of the Gaussian process regression model of the task-parametrised and the GMM parameters are automatically optimised.
3. Derived from the Bayesian Committee Machine, the prediction of the new task trajectories is derived from the weight contributions of all the trained Gaussian process regression models from corresponding movement-primitive datasets.



**Fig. 1** The overview of the proposed distributed probabilistic framework. This framework aims to accommodate various movement primitives in an overall framework and generate a novel movement

4. In order to demonstrate the application feasibility of our proposed distributed probabilistic framework, the task-parametrised GMM methodology is compared with our proposed distributed framework. Moreover, the application feasibility of this framework is further verified through real-world experiments.

### 3 Individual movement primitive learning

We start Sect. 3.1 by briefly introducing the learning process of encoding human demonstrations with GMM clustering and retrieving trajectories using GMR regression [3]. Moreover, the model learning of the movement primitives with the EM algorithm is given in Sect. 3.2.

#### 3.1 Human demonstration encoding

Basically, the  $i$ -th human demonstration can be defined as a dataset  $\{\xi^I, \xi^O\}_i$ , where  $\xi^I \in \mathbf{R}^I$  is considered as an time input variable. Hence,  $\xi^O \in \mathbf{R}^O$  is hence in either task space or joint space. Encoded by a GMM with  $K$  Gaussian processes, a datapoint  $\xi = [\xi^I, \xi^O]$  of  $D$  dimensions described by the GMM can be probabilistically defined as

$$p(\xi) = \sum_{k=1}^K \pi_k \mathcal{N}(\xi; \mu_k, \Sigma_k),$$

with the Gaussian distribution

primitive by probabilistically combining several primitives together with Bayesian Committee Machine

$$\mathcal{N}(\xi; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^D |\Sigma_k|} \exp^{-\frac{1}{2}(\xi - \mu_k)^T \Sigma_k^{-1} (\xi - \mu_k)},$$

where  $\mu_k$  and  $\Sigma_k$  are the mean and covariance of the Gaussian distribution  $\mathcal{N}(\xi; \mu_k, \Sigma_k)$  and  $\pi_k$  ( $\sum_k \pi_k = 1$ ) is the prior. Considering the input and output components separately

$$\mu_k = \begin{bmatrix} \mu_k^I \\ \mu_k^O \end{bmatrix}, \Sigma_k = \begin{bmatrix} \Sigma_k^I & \Sigma_k^{IO} \\ \Sigma_k^{OI} & \Sigma_k^O \end{bmatrix},$$

the predicted distribution  $p(\xi^O | \xi^I, k) \sim \mathcal{N}(\hat{\xi}_k, \hat{\Sigma}_k)$  is defined as

$$\hat{\xi}_k = \mu_k^O + \Sigma_k^{OI} (\Sigma_k^I)^{-1} (\xi^I - \mu_k^I),$$

$$\hat{\Sigma}_k = \Sigma_k^O - \Sigma_k^{OI} (\Sigma_k^I)^{-1} \Sigma_k^{IO}.$$

If we take the complete GMM into consideration, the predicted distribution can be rewritten as

$$p(\xi^O | \xi^I) \sim \sum_{k=1}^K h_k \mathcal{N}(\hat{\xi}_k, \hat{\Sigma}_k), \tag{1}$$

where  $h_k$  is the posterior that decides the responsibility of the  $k$ -th Gaussian distribution

$$h_k = \frac{p(k)p(\xi^I | k)}{\sum_{j=1}^K p(k)p(\xi^I | k)} = \frac{\pi_k \mathcal{N}(\xi^I; \mu_k^I, \Sigma_k^I)}{\sum_{j=1}^K \pi_j \mathcal{N}(\xi^I; \mu_j^I, \Sigma_j^I)}. \tag{2}$$

According to the linear combination properties of the Gaussian distributions, the conditional distribution is thus

estimated as a single Gaussian distribution. Given  $\xi^l$ , the expectation and covariance of  $\xi^O$  are approximated as

$$\hat{\xi} = \sum_{k=1}^K h_k \hat{\xi}_k, \quad \hat{\Sigma} = \sum_{k=1}^K h_k^2 \hat{\Sigma}_k. \tag{3}$$

### 3.2 Model learning with the EM algorithm

We utilise the EM (Expectation Maximisation) algorithm [16] for GMM training, which is an iterative algorithm to maximise the posterior estimation of parameters in the statistic model. According to Jensen’s inequality and KL (Kullback–Leibler) divergence, the EM algorithm consists of two steps, i.e. expectation step and maximisation step.

If we define the maximisation parameters of the GMM model as  $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ , the expectation step is trying to find the value of the following object function at  $g$  step

$$\begin{aligned} Q(\Theta^{(g)}) &= \sum_{k=1}^K \sum_{i=1}^N \ln(\pi_k) p(k|\mathbf{x}_i, \Theta^{(g)}) \\ &+ \sum_{k=1}^K \sum_{i=1}^N \ln[\mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k)] p(k|\mathbf{x}_i, \Theta^{(g)}), \end{aligned} \tag{4}$$

with  $x_i$  the training data. In the maximisation step, the parameter  $\Theta^{(g+1)}$  is thus obtained by maximising  $Q(\Theta^{(g)})$

$$\pi_k^{(g+1)} = \frac{1}{N} \sum_{i=1}^N p(k|\mathbf{x}_i, \Theta^{(g)}), \tag{5}$$

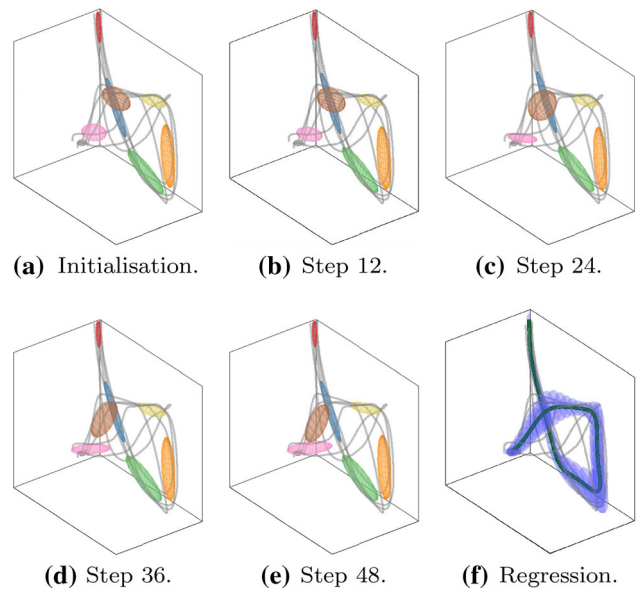
$$\mu_k^{(g+1)} = \frac{\sum_{i=1}^N \mathbf{x}_i p(k|\mathbf{x}_i, \Theta^{(g)})}{\sum_{i=1}^N p(k|\mathbf{x}_i, \Theta^{(g)})}, \tag{6}$$

$$\Sigma_k^{(g+1)} = \frac{\sum_{i=1}^N [\mathbf{x}_i - \mu_k^{(g+1)}][\mathbf{x}_i - \mu_k^{(g+1)}]^T p(k|\mathbf{x}_i, \Theta^{(g)})}{\sum_{i=1}^N p(k|\mathbf{x}_i, \Theta^{(g)})}. \tag{7}$$

For initialisation, the K-means algorithm [8] is utilised to choose the original parameters of the Gaussian distributions and hence the EM algorithm proceeds until converging and deriving a closed-form solution. Also, the graphic explanation of encoding the human demonstrations is given in Fig. 2.

### 4 Probabilistic distributed framework

According to the analysis of individual movement primitive in Sect. 3, a learned individual primitive model based on several human demonstrations can be represented by GMM parameters  $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ . Inspired by [2], if a connection between the GMM parameters and the task-



**Fig. 2** The graphic explanation of encoding human demonstrations. In the initialisation step, seven Gaussian distribution models are initialised with K-means. After 48 steps training with EM algorithm, the expectation of GMM converges to predefined interval. Therefore, the trajectory is hence retrieved using GMR as shown above

specific feature is established, an individual primitive model could generate more extension. Therefore, we introduce a Gaussian process regression model that maps the Cartesian task parameters to GMM parameters in Sect. 4.1. Also, the probabilistic distributed learning framework for multiple movement primitives is detailed in Sect. 4.2.

#### 4.1 Task-parametrised model

In order to encode the relationship between the task parameter  $\mathcal{Q}$  and the GMM parameters  $\Theta$ , we consider a regression model based on Gaussian process

$$\Theta = f(\mathcal{Q}) + \omega, \quad \omega \sim \mathcal{N}(0, \Sigma_\omega), \tag{8}$$

with the Gaussian white noise  $\omega$  and the variance  $\Sigma_\omega$ .

The regression model can be fully specified by the mean function  $m_f(\cdot)$  and semi-positive covariance function  $k_f(\cdot, \cdot)$ . Moreover, the kernel covariance is defined as

$$k(Q_i, Q_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(Q_i - Q_j)\Lambda^{-1}(Q_i - Q_j)^T\right) \delta_{ij} + \sigma_\omega^2, \tag{9}$$

with the length-scales  $\Lambda = \text{diag}(l_1^2, \dots, l_n^2)$ , the signal variance  $\sigma_f^2$ , and the noise variance  $\sigma_\omega^2$ , which are defined as the GP hyper-parameters  $\theta = \{l_i, \sigma_f, \sigma_\omega\}$ .

Given desired task parameters  $\mathcal{Q}^d$ , the new GMM parameters derived from conditional probability of the Gaussian distribution are defined as

$$m_f(Q^d) = k_*^T (K + \sigma_\omega^2 I)^{-1} y, \tag{10}$$

$$k_f(Q^d, Q^d) = k_{**} - k_*^T (K + \sigma_\omega^2 I)^{-1} k_*, \tag{11}$$

where  $k_* = k(Q, Q^d)$  and  $k_{**} = k(Q^d, Q^d)$ .

In [2], the covariance  $k_f(Q^d, Q^d)$  of conditional probability is neglected and only the mean  $m_f(Q^d)$  is used to retrieve novel trajectories. As the hyper-parameters are not optimised, the covariance may have a negative value. In the proposed framework, the covariance  $k_f(Q^d, Q^d)$  is seen as the crucial information that is utilised to indicate the confidence interval in the data fusion. The hyper-parameters of the Gaussian process model should be optimised, and therefore, the covariance could have a meaningful value which indicates a positive connection among different GMM parameters.

After choosing a flat  $p(\theta)$ , the posterior distribution is only proportional to the marginal likelihood

$$\begin{aligned} p(\Theta|Q) &= \int \int p(\Theta|Q, f, \theta) p(f|\theta) p(\theta) df d\theta, \\ &= \int p(\Theta|Q, \theta) p(\theta) d\theta. \end{aligned}$$

To optimise the vector of hyper-parameters  $\theta$ , we follow the recommendation from [13]. Particularly, the log-marginal likelihood can be given as

$$\begin{aligned} \log p(\Theta|Q, f, \theta) &= \log \int p(\Theta|Q, f, \theta) p(f|\theta) df \\ &= -\frac{1}{2} \Theta^T (K + \sigma_\omega^2 I)^{-1} \Theta - \frac{1}{2} \log |K + \sigma_\omega^2 I| - \frac{D}{2} \log(2\pi) \end{aligned} \tag{12}$$

Therefore, the hyper-parameters are set by maximising the marginal likelihood. We define the partial derivatives of the marginal likelihood w.r.t. the hyper-parameters  $\theta_i$  [26]

$$\frac{\partial}{\partial \theta_i} \log p(\Theta|Q, \theta) = \frac{1}{2} \Theta^T K_\sigma^{-1} \frac{\partial K}{\partial \theta_i} K_\sigma^{-1} \Theta - \frac{1}{2} \text{tr} \left( K_\sigma^{-1} \frac{\partial K_\sigma}{\partial \theta_i} \right). \tag{13}$$

where  $K_\sigma = K + \sigma_\omega^2 I$ . In the above equation, the two terms usually refer to the data-fit term and the model complexity. The gradient technique aims to seek the trade-off between the data-fit and model complexity.

### 4.2 Distributed learning

The obtained covariance  $k_f(Q^d, Q^d)$  of Gaussian distribution shows the confidence interval of the predictions, which could be seen as the robustness of Gaussian process regression. In this paper, the covariance of prediction is used as a data fusion indicator.

Owing to independence assumption, the marginal likelihood could be factorised into several individual terms

$$p(\Theta|Q, \theta) = \prod_{k=1}^M p_k(\Theta^{(k)}|A^{(k)}, \theta), \tag{14}$$

where each factor term  $p_k$  depends on the  $k$ -th individual GP regression model as discussed in sect. 4.1.

The following information details how to combine  $M$  individual primitive models to form an overall prediction with the Bayesian Committee Machine (BCM) [23]. As we can see, the BCM explicitly combines the GP prior  $p(f)$  when making prediction.

Given  $M$  individual primitive models, the predictive distribution can be generally defined by

$$p(f_*|D^{(1)}, \dots, D^{(M)}) \propto p(D^{(1)}, \dots, D^{(M)}|f_*) p(f_*), \tag{15}$$

with  $p(f_*)$  the prior over functions and  $D^{(k)}, k = 1, \dots, M$  the  $k$ -th dataset. Under BCM conditional independence assumption, the predictive is rewritten as

$$p(f_*|D^{(1)}, \dots, D^{(M)}) \propto p(f_*) \prod_k p(D^{(k)}|f_*), \tag{16}$$

$$= \frac{\prod_{k=1}^M p(D^{(k)}, f_*)}{p^{M-1}(f_*)} \tag{17}$$

$$\propto \frac{\prod_{k=1}^M p_k(f_*|D^{(k)})}{p^{M-1}(f_*)}. \tag{18}$$

Therefore, given an input  $x_*$ , the posterior predictive distribution is defined as

$$p(f_*|x_*, D) = \frac{\prod_{k=1}^M p_k(f_*|x_*, D^{(k)})}{p^{M-1}(f_*)|x_*}, \tag{19}$$

Then the mean and the precision are

$$\mu_* = (\sigma_*)^2 \sum_{k=1}^M \sigma_k^{-2}(x_*) \mu_k(x_*) \tag{20}$$

$$(\sigma_*)^{-2} = \sum_{k=1}^M \sigma_k^{-2}(x_*) + (1 - M) \sigma_{**}^{-2} \tag{21}$$

separately, with  $\sigma_{**}^{-2}$  the prior covariance of  $p(f_*)$ .

## 5 Experiments

In order to verify the feasibility of the proposed probabilistic framework, several experiments are implemented in this section. In Sect. 5.1, the task-parametrised GMM is compared in terms of several aspects, such as generalisation capability, accuracy and computation expense. Furthermore, two assembly tasks, rivets and nutplate pick-up



are given in Sect. 5.2 to demonstrate real-world application feasibility.

### 5.1 Comparison with task-parametrised GMM

The task-parametrised GMM is a powerful tool to retrieve trajectories in a variety of tasks, such as movement primitive reproduction, viapoint adaptation and modulation. The proposed probabilistic distributed framework aims to mutually combine and simultaneously accommodate various movement primitives in an overall scenario, and therefore augments the generalisation capability and makes great use of every single movement primitive. In this subsection, we would like to explore more functions both from task-parametrised GMM and our proposed framework in terms of generalisation capability, accuracy and robustness, and computation expense.

*Generalisation capability* for exploring the generalisation capability, twelve movement primitive datasets are generated randomly as shown in Fig. 3. Each dataset accommodates four movement primitives with three GMM components. Moreover, the origin frame and task frame are recorded in pink and green separately for further analysis. It is worth pointing out that basically every dataset could be seen as a task-parametrised GMM model.

Four different task frames are presented for testing the generalisation capability, as shown in Fig. 4. Particularly, for a desired task frame in green, every dataset retrieves its own predicted trajectory as given in Fig. 4a–d. Moreover, three GMM components are displayed with the mean in black dot and the covariance in blue, yellow and purple ellipses.

As shown in Fig. 4, although all the movement datasets give their predictions, some of these predictions do not match the desired task frame in terms of position and orientation. This is because a single movement dataset has a limited generalisation capability. If the desired task frame

is too far from the task frames of the data sample, the task-parametrised will have a poor retrieving performance.

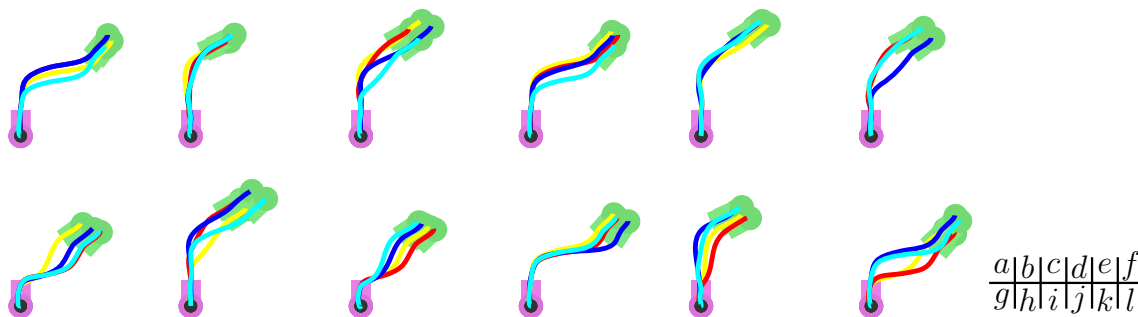
Our proposed distributed framework takes all the predictions from the datasets into consideration and fuses the trajectories on a GMM-parameter level. In addition, this probabilistic framework could bear poor prediction derived from several datasets, and meanwhile, output satisfying results as presented in Fig. 5.

*Accuracy and robustness* in order to provide a more comprehensive analysis, the weights and prediction intervals of every dataset are presented in Fig. 6 derived from Eqs. 20 and 21, separately. Moreover, the prediction accuracies of each primitive dataset and our proposed distributed framework are compared in Fig. 7.

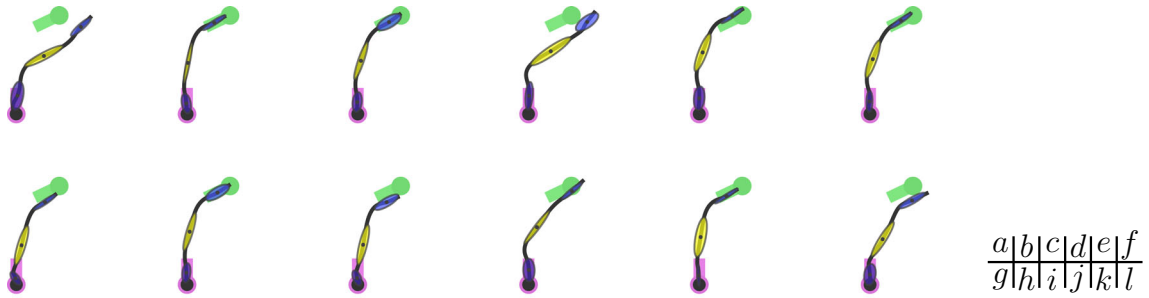
As shown in Fig. 6, the confidential interval in the red bar shows the prediction range of each movement dataset. If the confidential interval is large, then the corresponding movement dataset will lose its confidence in predicting novel trajectories. On the contrary, if the confidential interval is narrow, then the movement dataset has more faith in its own prediction. Consequently, in our proposed distributed framework, large confidential interval matches with low weight as presented in Fig. 6 in blue and vice versa.

As shown in Fig. 7a, the prediction error of each primitive dataset is nearly proportional to the confidential intervals in Fig. 6a. The similar situations can also be observed in the other three group simulations, i.e. Figs. 7b and 6b, Figs. 7c and 6c, and Figs. 7d and 6d. This is why we use the information of the confidential intervals of each primitive dataset are used to quantitatively explain the weights applied in Equ. 20. In addition, the proposed distributed framework shown in green gives a better prediction accuracy compared with the accuracy from each primitive datasets shown in yellow according to the four prediction errors given in Fig. 7a–d.

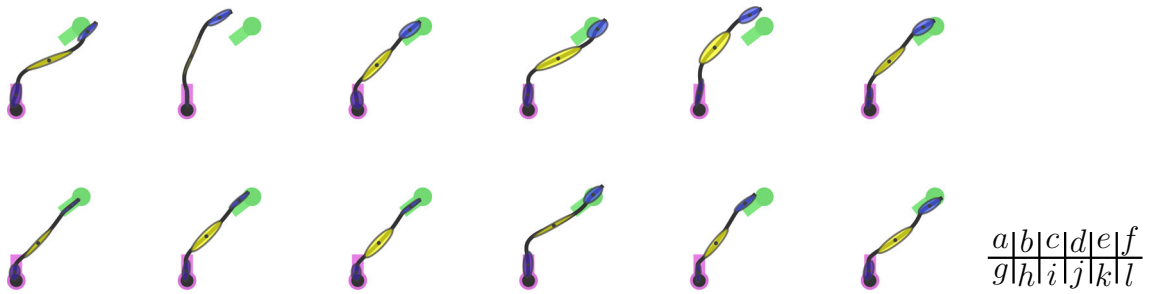
We would like to point out that it is not always the case that a very small confidential interval leads to better



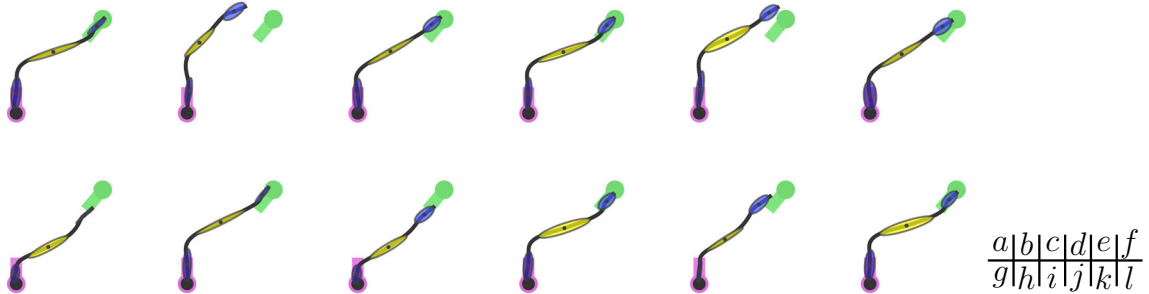
**Fig. 3** Twelve movement-primitive datasets. Each dataset generated randomly has four movement primitives. The task frame and origin frame contain the information of position and orientation shown in green and pink, respectively (color figure online)



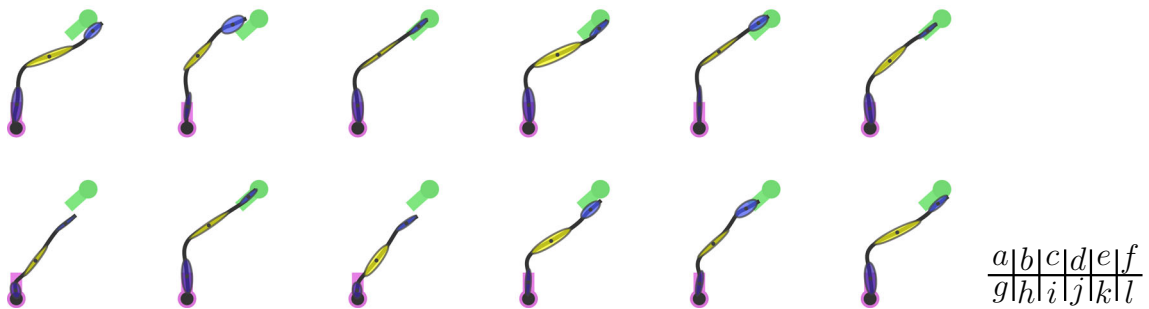
(a) Task frame (green) and the retrieved of the first group primitive datasets.



(b) Task frame (green) and the retrieved of the second group primitive datasets.

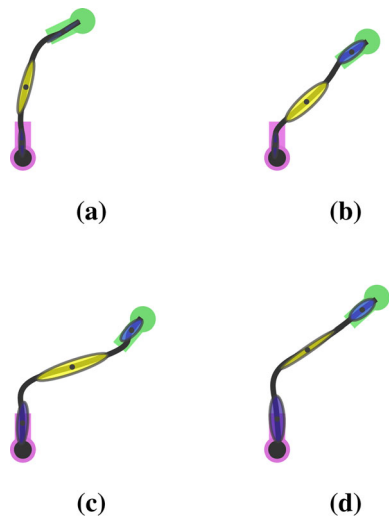


(c) Task frame (green) and the retrieved of the third group primitive datasets.

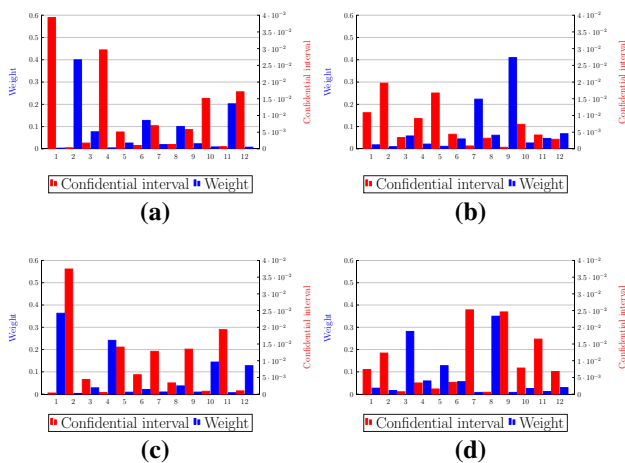


(d) Task frame (green) and the retrieved of the forth group primitive datasets.

**Fig. 4** The retrieved trajectories of movement-primitive datasets. The generalisation capability is tested with four different task frames given in green. The initial frame is shown in pink, and three GMM components corresponding to each movement primitive are presented in blue, yellow and purple. In addition, each retrieved trajectory could be seen as the prediction of the task-parametrised GMM (color figure online)

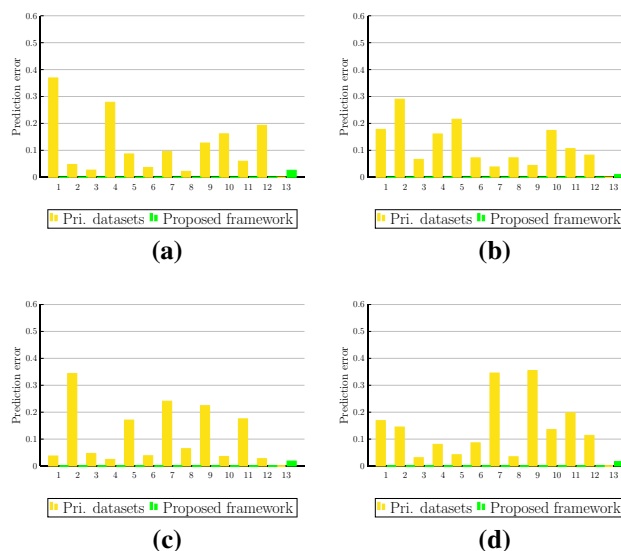


**Fig. 5** The retrieved trajectories of our proposed distributed framework. For a group of twelve movement primitive datasets given in Fig. 4a–d, our proposed framework could accommodate all the primitive datasets together and predict novel trajectories regarding the desired task frame



**Fig. 6** The confidential interval and weights corresponding to each movement primitive dataset in Fig. 4

prediction results. Sometimes, a narrow confidential interval indicates that the algorithm is very aggressive. Additionally, a large confidential interval may result in conservative predictions. So it is crucial to keep a balance between uncertainty and over-fitting and maintain robustness.



**Fig. 7** The comparison of the prediction accuracies. The prediction accuracies corresponding to four groups primitive datasets are shown in yellow, along with the prediction accuracy of the proposed framework given in green (color figure online)

*Computation expense* another crucial property we add to the distributed framework is the optimisation of the hyper-parameters of the Gaussian process with Evidence Maximisation. However, the training of hyper-parameter requires additional computation expense  $\mathcal{O}(n^3)$  and  $\mathcal{O}(n^2)$  for prediction if the trained parameters are cached, with  $n$  the volume of the training dataset. Therefore, for our proposed framework, the whole computation expense is  $\mathcal{O}(m * n^3 + m * n^2)$ , with  $m$  primitive datasets. For more information on the computation expense of the distributed framework, we refer to our previous work [25].

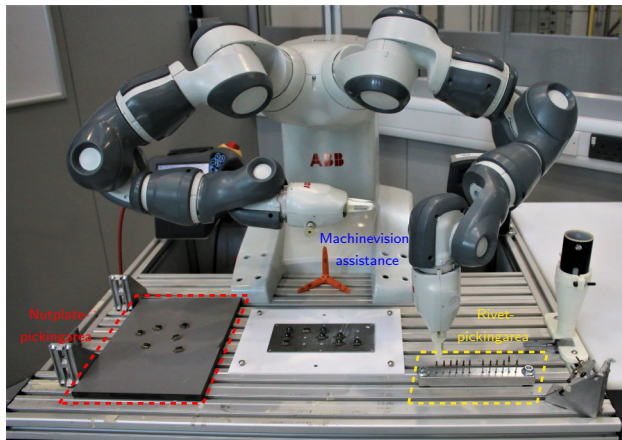
### 5.2 Assembly tasks

After addressing all the key issues of our proposed distributed framework in Sect. 5.1, in this subsection, the feasibility with real-world experiments is verified, such as rivet picking and nutplate picking. As presented in Fig. 9, we test our proposed framework with the ABB YuMi robot. The YuMi is a two-arm collaborative robot with an industrial camera mounted on the wrist of the right arm, and the payload is 0.5 kg for each arm. To amplify the function of the YuMi, two grippers are equipped with two arms, respectively. *Rivet picking* the first experiment implemented in this subsection is picking rivet from the rivet block as shown in Fig. 10. We collect twelve groups of human demonstrations as given in Fig. 8, along with the trajectories in Fig. 11a. As shown in Fig. 11a, the collected demonstrations have some inaccuracies. Particularly, the demonstrations are not smooth enough and some of them

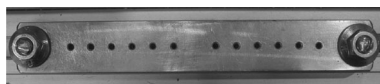




**Fig. 8** Human demonstrations of rivet picking. During the human demonstrations, the YuMi robot is in lead-through model and the trajectories are captured using socket communication to an external PC



**Fig. 9** The experimental platform. We test our proposed distributed framework with the ABB collaborative robot, YuMi. The rivet block is on the left side of the YuMi, while the picking board is located on the right side



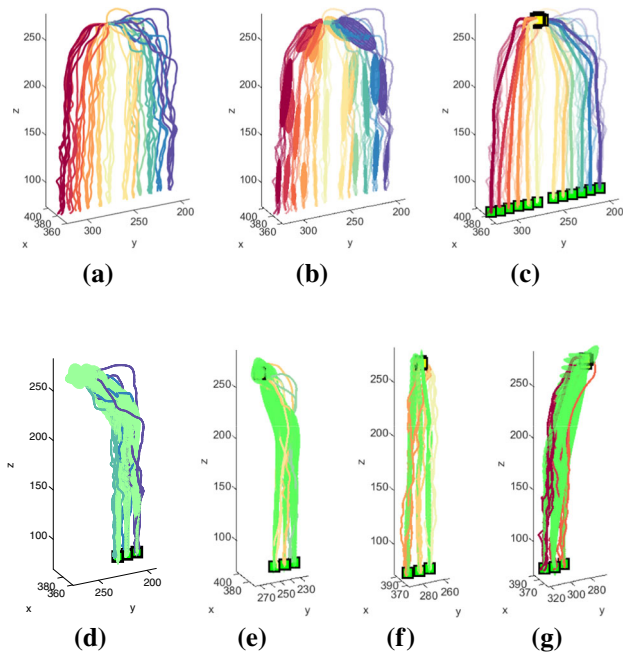
**Fig. 10** The top view of the rivet block. The rivet block is designed to locate the rivets. In addition, the diameter of each hole is 3 mm

may not be successfully inserted into the holes of the rivet block.

Each primitive-dataset group is trained with three GMM using EM algorithm as presented in Fig. 11b. Moreover, as shown in Fig. 11b if the training dataset is decentralised, the GMM ellipsoid is large and the Gaussian process model has a wide distribution and verse versa.

Besides, we construct a Gaussian process regression between the task frames of twelve primitive datasets and corresponding GMM model parameters. Under our proposed distributed framework, the twelve novel trajectories are inferred in Fig. 11c, along with the desired task frames in square black and green makers and the origin frames in square black and yellow makers.

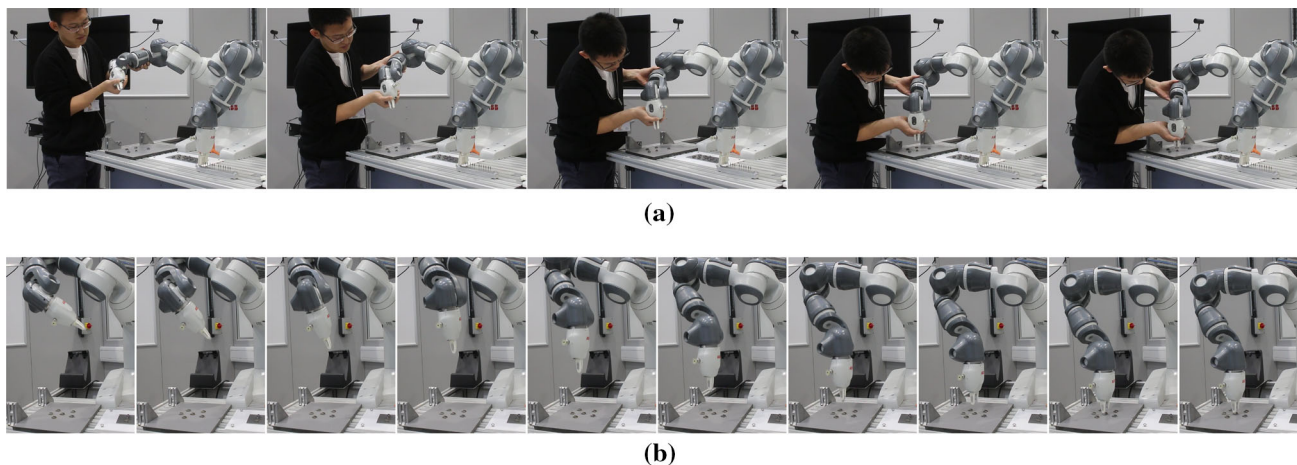
To reveal further details, each three consecutive predicted trajectories are separated in four figures as



**Fig. 11** The learning process of our proposed probabilistic distributed framework. The training trajectories are collected in (a). Then, the trained GMM components of each dataset using the EM algorithm are given in (b). The retrieved trajectories inferred by twelve movement primitive datasets are shown in (c) and the desired frames are given in green markers. Finally, the confidential interval of each three consecutive trajectories according to GMR is given in (d), (e), (f) and (g)

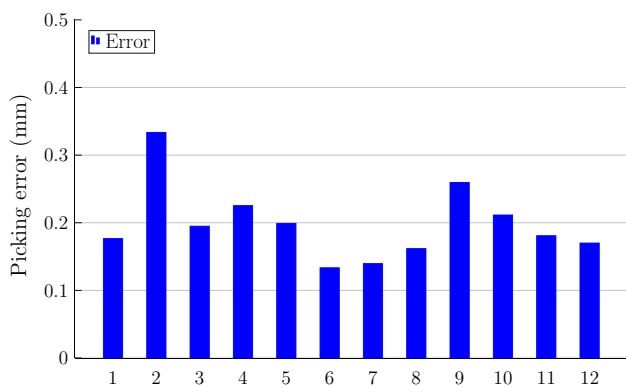
represented in Fig.11d–g. Additionally, the confidential intervals derived from GMR are plotted in green.

The prediction errors are given in Fig. 13. As we can see in the figure, all the prediction errors are below 0.35 mm. Most of the prediction errors are lower than 0.2 mm, which is the reference assembly precision of aerospace manufacturing. However, we still notice that the prediction error of the ninth hole is larger than 0.2 mm and the second is even higher than 0.3 mm. This is mainly caused by the accuracy of the human demonstrations. If all the human demonstrations are far from the desired target, the prediction will have poor retrieving performance. We would like to point out that the above precision or prediction errors are enough accurate for picking applications, such as rivet



**Fig. 12** Human demonstrations of the nutplate picking and the record of the nutplate picking process. **a** Human demonstrations of nutplate picking. **b** Robot motion record of the nutplate picking. In this

experiment, the setting is the same as rivet picking. However, the location of the nutplate picking is obtained with machine vision



**Fig. 13** The prediction errors of the rivet picking experiments

picking. Theoretically, the proposed distributed framework can manage to keep the prediction errors below 0.2 mm with more accurate demonstrations.

*Nutplate picking* the second experiment is nutplate picking, which is implemented with machine vision techniques using Cognex in-Sight smart camera. The initial experiment setting includes the checkerboard calibration to the YuMi robot and features extractions, which is achieved with the function PatMax Patterns [1]. Besides, we set an adapted exposure time which depends on the ambient lighting condition.

Similarly, we obtain several human demonstrations as presented in Fig. 12a. Then, these demonstrations are trained with GMM models and retrieve novel trajectories under the proposed distributed probabilistic framework.

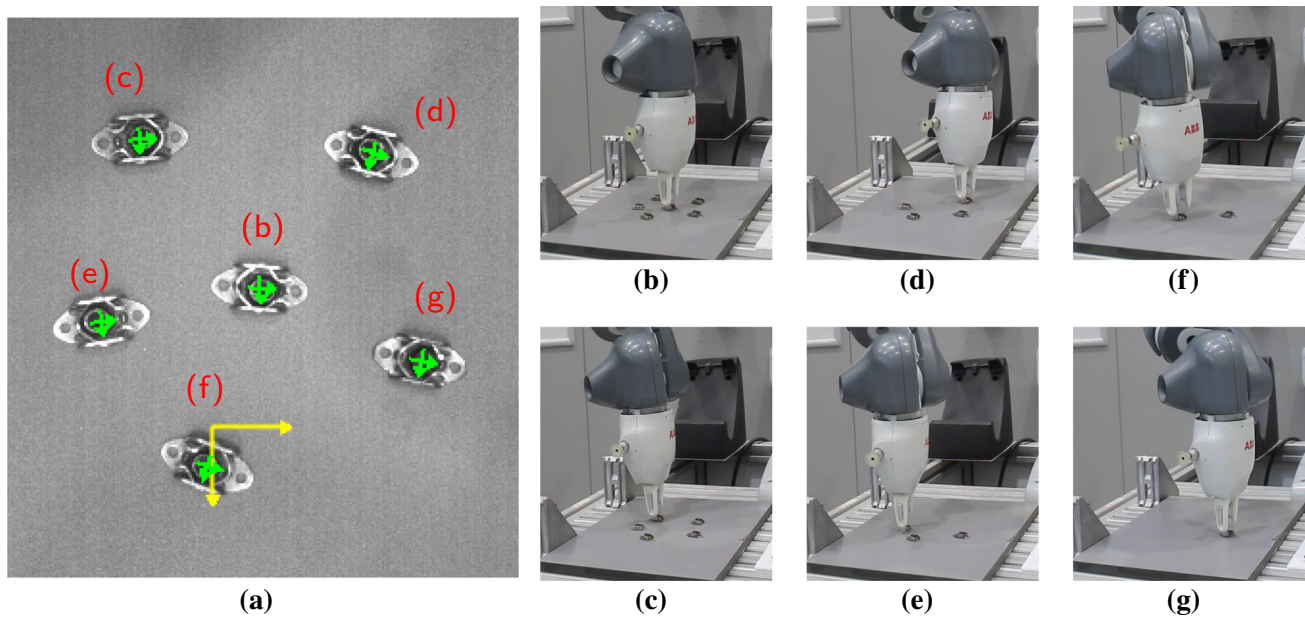
For the picking guidance, the position and orientation of the nutplates are located using the machine vision techniques, as given in Table 1. As given in the left subfigure in Fig. 14, the positions of each nutplate are showing in the

**Table 1** Extracted coordinates using machine vision

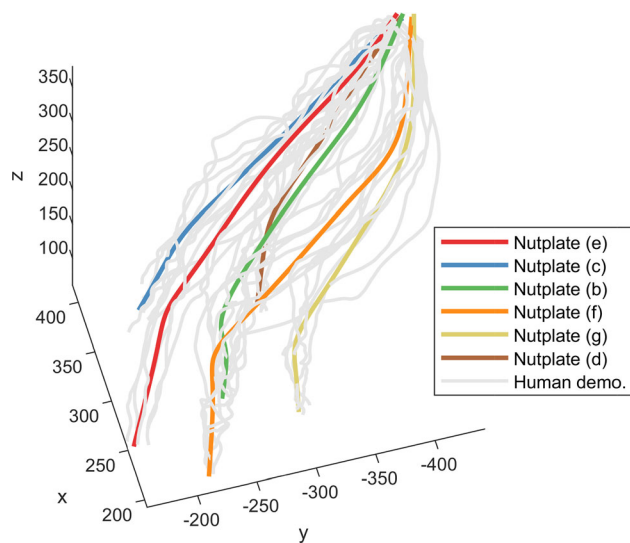
Coordinate	$u$	$v$	$\alpha$
Point 1	- 34.465	12.137	- 4.035
Point 2	- 71.713	- 1.833	6.909
Point 3	9.729	- 1.043	- 19.725
Point 4	- 26.218	- 27.360	10.943
Point 5	- 68.619	40.482	- 13.712
Point 6	- 16.294	52.771	- 11.322

green cross, and the orientations are presented with arrows. The picking positions and orientations of the gripper are presented in the right six subfigures in Fig. 14. The additional information of the learned trajectories is given in Fig. 15. Besides, the picking process of the human demonstration and the nutplate in the middle is recorded in Fig. 12a and b, respectively. We would like to point out that only the target positions are learned by the proposed distributed framework and the orientations are directly sent to the picking program. The orientation learning will be addressed in our future work. In order to further analyse the picking accuracy, several experiments are implemented with the same experimental setting.

Combining these nutplate picking experiments, the average prediction errors with error bars are presented in Fig. 16. The average errors are below 0.3 mm, which include implementation error, machine vision error, and the error of our proposed framework. The machine vision error is derived from the lens distortion and ambient lighting condition. With the compensation of the gripper, the YuMi can successfully perform nutplate picking tasks.



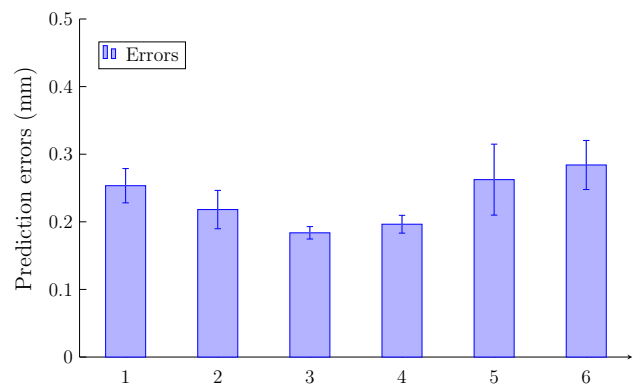
**Fig. 14** The desired task frames and gripper picking records. The desired task frame is derived from machine vision corresponding to the information of Table 1. The gripper picking actions are recorded at the right side of the figure



**Fig. 15** The human demonstration and learned trajectories with the proposed probabilistic framework. This figure provides further information for Fig. 14. The human demonstration is presented in grey colour. Additionally, the learned trajectories are given in colour solid lines

## 6 Conclusion

In this paper, we propose a novel distributed probabilistic framework, which can accommodate various movement primitives together and retrieve novel trajectories in a weight-based scenario. Specifically, the core idea of this framework is to not only provide functionalities of



**Fig. 16** The prediction errors of the nutplates picking

generating new movement primitive given task parameters but also aim to explore a feasible solution to save various primitives and select or modulate them regarding different demands.

The human demonstration for establishing the primitive dataset is captured with GMM and GMR. Moreover, the regression model between the task parameters and primitives parameters is obtained by the Gaussian process and could be automatically optimised with Evidence Maximisation. Also, given the desired task frame, the retrieved trajectories are predicted using Bayesian Committee Machine. The assembly task experiments, such as rivet and nutplate picking, show the application feasibility of our proposed framework. Our future work will focus on the



movement primitives library as well as the enhancement of precision.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. ABB A (2015) Robotics products. Application manual-integrated vision
2. Calinon S (2016) A tutorial on task-parameterized movement learning and retrieval. *Intel Serv Robot* 9(1):1–29
3. Calinon S, Guenter F, Billard A (2007) On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans Syst, Man, Cybern, Part B (Cybern)* 37(2):286–298
4. Cohn DA, Ghahramani Z, Jordan MI (1996) Active learning with statistical models. *J Artif Intell Res* 4:129–145
5. Huang Y, Rozo L, Silvério J, Caldwell DG (2019) Kernelized movement primitives. *Int J Robot Res* 38(7):833–852
6. Jaquier N, Ginsbourger D, Calinon S (2019) Learning from demonstration with model-based gaussian process. *arXiv preprint arXiv:191005005*
7. Kramberger A, Gams A, Nemeč B, Chrysostomou D, Madsen O, Ude A (2017) Generalization of orientation trajectories and force-torque profiles for robotic assembly. *Robot Auton Syst* 98:333–346
8. Krishna K, Murty MN (1999) Genetic k-means algorithm. *IEEE Trans Syst, Man, Cybern, Part B (Cybern)* 29(3):433–439
9. Kulvicius T, Biehl M, Aein MJ, Tamosiunaite M, Wörgötter F (2013) Interaction learning for dynamic movement primitives used in cooperative robotic tasks. *Robot Auton Syst* 61(12):1450–1459
10. Lambrecht J, Kleinsorge M, Rosenstrauch M, Krüger J (2013) Spatial programming for industrial robots through task demonstration. *Int J Adv Rob Syst* 10(5):254
11. Laursen JS, Schultz UP, Ellekilde LP (2015) Automatic error recovery in robot assembly operations using reverse execution. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 1785–1792
12. Laursen JS, Ellekilde LP, Schultz UP (2018) Modelling reversible execution of robotic assembly. *Robotica* 36(5):625–654
13. MacKay DJ (1999) Comparison of approximate methods for handling hyperparameters. *Neural Comput* 11(5):1035–1068
14. Maeda G, Ewerton M, Lioutikov R, Amor HB, Peters J, Neumann G (2014) Learning interaction for collaborative tasks with probabilistic movement primitives. In: 2014 IEEE-RAS International Conference on Humanoid Robots, IEEE, pp 527–534
15. Meirbek M, Meifa H, Zhemin T (2020) Current issues in uncertainty of dimensional tolerance metrology and the future development in the domain of tolerancing. In: IOP Conference Series: Materials Science and Engineering, IOP Publishing, vol 715, p 012084
16. Moon TK (1996) The expectation-maximization algorithm. *IEEE Signal Process Mag* 13(6):47–60
17. Paraschos A, Daniel C, Peters JR, Neumann G (2013) Probabilistic movement primitives. In: Advances in neural information processing systems, pp 2616–2624
18. Peternel L, Petrič T, Babič J (2018) Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation. *Auton Robot* 42(1):1–17
19. Peters J, Schaal S (2008) Natural actor-critic. *Neurocomputing* 71(7–9):1180–1190
20. Schaal S (2006) Dynamic movement primitives—a framework for motor control in humans and humanoid robotics. In: Adaptive motion of animals and machines, Springer, pp 261–280
21. Stulp F, Sigaud O (2012) Policy improvement methods: Between black-box optimization and episodic reinforcement learning
22. Theodorou E, Buchli J, Schaal S (2010) A generalized path integral control approach to reinforcement learning. *J Machine Learn Res* 11:3137–3181
23. Tresp V (2000) A bayesian committee machine. *Neural Comput* 12(11):2719–2741
24. Wan W, Harada K (2016) Integrated assembly and motion planning using regrasp graphs. *Rob Biomim* 3(1):1–11
25. Wang L, Du Z, Dong W, Shen Y, Zhao G (2018) Probabilistic sensitivity amplification control for lower extremity exoskeleton. *Appl Sci* 8(4):525
26. Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning, vol 2. MIT press Cambridge, MA
27. Xu J, Hou Z, Wang W, Xu B, Zhang K, Chen K (2018) Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks. *IEEE Trans Industr Inf* 15(3):1658–1667

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.