IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2024)

Learning Multi-Reference Frame Skills from Demonstration with Task-Parameterized Gaussian Processes

Mariano Ramírez Montero*, Giovanni Franzese*, Jens Kober and Cosimo Della Santina

***Abstract*— A central challenge in Learning from Demonstration is to generate representations that are adaptable and can generalize to unseen situations. This work proposes to learn such a representation without using task-specific heuristics within the context of multi-reference frame skill learning by superimposing local skills in the global frame. Local policies are first learned by fitting the relative skills with respect to each frame using Gaussian Processes (GPs). Then, another GP, which determines the relevance of each frame for every time step, is trained in a self-supervised manner from a different batch of demonstrations. The uncertainty quantification capability of GPs is exploited to stabilize the local policies and to train the frame relevance in a fully Bayesian way. We validate the method through a dataset of multi-frame tasks generated in simulation and on real-world experiments with a robotic manipulation pick-and-place re-shelving task.**

We evaluate the performance of our method with two metrics: how close the generated trajectories get to each of the task goals and the deviation between these trajectories and test expert trajectories. According to both of these metrics, the proposed method consistently outperforms the state-of-the-art baseline, Task-Parameterised Gaussian Mixture Model (TPGMM).

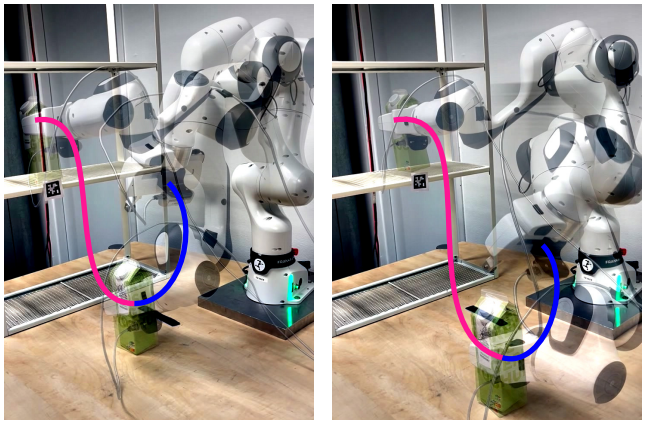


Fig. 1. In this work, we propose TPGP, a GP-based architecture to learn by demonstration tasks composed of multiple sub-tasks, like grasping and re-shelving objects. The two panels show two such executions.

I. Introduction

As robots become more ubiquitous in our society, it is necessary to easily teach them flexible skills on the fly. A promising possibility is to use Learning from Demonstration [1] to transfer knowledge and skills to the robot since this can allow easy programming of robots, even for non-roboticists [2]. Although different regressors such as Gaussian Mixture Models [3], Neural Networks [4], Gaussian Processes [5], and Dynamic Movement Primitives [6] have been proposed to learn skills with respect to the global or object frames, determining frame relevance at each time step is still an open challenge.

For example, Fig. I, illustrates a re-shelving task. Here, it is natural to consider this task as composed of two parts, grasping and shelving, one requiring the robot to reproduce movements with respect to the object frame and the other to the desired shelf location frame. While associating the right frame w.r.t. which to perform generalization may appear trivial to a human expert, it is much less so for a robot learner. This challenge has been formalized in the context of Learning from Demonstration in [2]. As we will discuss more in detail in Sec. II, this challenge is often solved by introducing ad hoc heuristics or by explicitly labeling to identify which frame is relevant at each time step. However, labeling is impractical when dealing with non-expert human teachers or with many frames, and task-specific heuristics are not flexible. A method to generate self-supervised relevance determination of each frame, directly inferred from the demonstrations, forgoes the need for labeling and increases flexibility.

For this reason, we propose Task-Parameterised Gaussian Processes (TPGP), a method for learning skills parameterized by given coordinate reference frames. This method contributes to the topic of teaching multi-reference frame skills without using any supervised segmentation algorithm or task-specific heuristics for the reference frame selection. At every time step, local policies output desirable movement relative to each frame while a relevance model selects the most likely frame. This method uses Gaussian processes to capture and reject uncertainties in the learned local dynamics and resolve ambiguities in weighing the frames' relevance.

The performance of TPGP is also compared to another self-supervised, heuristic- and segmentation-free state-of-the-art method, Task-parameterized Gaussian Mixture Models (TPGMM) [7], [8]. In this comparison, we show better performance of TPGP through a metric that quantifies how close the generated trajectories get to the goals and through a metric that quantifies the trajectories' deviations from test expert trajectories.

The work by Mariano Ramírez Montero was supported under the European Union's Horizon Europe Program from Project EMERGE - Grant Agreement No. 101070918. Giovanni Franzese was supported by a contribution from the National Growth Fund program NXTGEN Hightech.

The authors are with Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: m.ramirezmontero-1@tudelft.nl, g.franzese@tudelft.nl, j.kober@tudelft.nl, c.dellasantina@tudelft.nl).

* denotes equal contribution

II. Related work

When learning a multi-reference frame policy, there are usually two approaches. The first is to segment the trajectory into many sub-motions and then find the most relevant frame for each of them. Alternatively, rather than relying on a segmentation algorithm, other algorithms solve the allocation problem for each time step in a continuous way.

In object manipulation tasks, segmentation of demonstrations often relies on changes in "contact relations" observed through end-effector distance to relevant objects or haptic signals. This object-centred approach aids generalization when the objects are in new configurations. For instance, [9] propose a hierarchical segmentation where they first segment based on changing contact relations, and then segment further based on acceleration profiles.

Another work analyses contact relations in human-provided demonstrations, considering hand-object relations again through proximity to objects, and additionally using hand velocity correlation to avoid false positives [10]. Segments generated with end-effector distance from an object can also be used as primitives in performing hierarchical reinforcement learning [11].

A threshold on the measured force magnitude at the end-effector and zero-velocity crossing events can be used as heuristics to perform segmentation [12]. Then, the relevant frame for each group of segments is obtained by selecting the frame that has the most consistent converging behaviour with respect to itself. Similarly, Directional Normal Distributions can be used as a way to measure the convergence and to group segments while also assigning frames to them [13].

While these methods show successful experimental validation, reliance on task-specific heuristics limits their flexibility. For tasks that are not necessarily object-centric, or even for objects of different dimensions if distance thresholds are used, these methods might not generalize well. The methods proposed in [14] and [15] exploit the variance in the data to identify important task constraints. Nonetheless, they still require an empirically set threshold to perform their variance-based segmentation. This is, again, task-specific and thus limits the flexibility of the method.

The approach in [7], [8], Task-parameterized Gaussian Mixture Models (TPGMM), is an exception since it does not use heuristics or pre-segmentation when learning multi-reference frame skills. The approach first transforms the demonstrated trajectories to all potentially relevant frames and then encodes each of these using Gaussian Mixture Models (GMMs). Then, in a new configuration, each of these Gaussians can be linearly transformed according to the new position of their corresponding frame. The resulting GMM for this situation can then be computed as the product of the transformed models, exploiting the fact that the product of two Gaussians is still a Gaussian.

Other non-probabilistic self-supervised frame relevance learning methods have been proposed in [16] and [17], where the relevance was obtained as a least-square optimization. However, [17] did not scale to more than two frames without the use of heuristics.

Since TPGMM is the only other comparable state-of-the-art method that is also probabilistic, heuristic-free and segmentation-free, we provide a comparison of the performance of TPGP and TPGMM. Through a metric that quantifies the deviation from the goals of a given task at each frame and the Fréchet distance between demonstrations and reproductions, we show that TPGP outperforms TPGMM.

III. Methodology

The core idea of our approach is to transform the demonstration data to the local reference frames to encode the relative dynamics, and then use a self-supervised approach to train a frame relevance predictor that selects the most relevant frame during execution. A diagram showing the main steps of the proposed method is shown in Fig. 2. Each of these steps is explained in detail in Section III-B, Section III-C, and Section III-D, respectively.

A. Demonstration Recording

During the recording of a demonstration, the 2D/3D position of the agent, e.g., the robot, is recorded. Each recorded position is augmented with a progress value φ between 0 and 1, calculated as the index of that datapoint divided by the total number of datapoints for that demonstration.

The state $\boldsymbol{x}$ of the system is thus composed of the Cartesian position $\boldsymbol{\xi}$ (2D or 3D) and the progress value φ, i.e., $\boldsymbol{x} := [\boldsymbol{\xi}, \varphi]$. A set of l sequential states $\boldsymbol{x}$ define a demonstration $\boldsymbol{d}_n = \{\boldsymbol{x}_{n,0}, \dots, \boldsymbol{x}_{n,l}\}$, where the subscript n denotes the n-th demonstration. The set of all given training demonstrations is then $\mathcal{D}^{(0)} = \{{}^0\boldsymbol{d}_0, \dots, {}^0\boldsymbol{d}_N\}$, where we add the superscript to indicate this is in the fixed, global (0th) frame, and N is the total number of demonstrations.

Moreover, before each demonstration, the positions and orientations of any relevant frames (relative to the fixed frame) are also recorded. For each frame m we can then construct a translation vector ${}^m\boldsymbol{t}$ and rotation matrix ${}^m\boldsymbol{R}$, and transform any datapoint ${}^0\boldsymbol{x}_{n,i}$ from frame 0 to frame m:

$$ {}^m\boldsymbol{H} = \begin{bmatrix} {}^m\boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{1} $$

$$ {}^m\boldsymbol{x}_{n,i} = \begin{bmatrix} {}^m\boldsymbol{t} \\ 0 \end{bmatrix} + {}^m\boldsymbol{H}\, {}^0\boldsymbol{x}_{n,i}. \tag{2} $$

Every demonstration ${}^0\boldsymbol{d}_n$ in $D^{(0)}$ can then be transformed, resulting in M transformed demonstration sets $\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(M)}$. Note that in the transformation operation, we do not need to rotate or translate the progress variable. These sets, plus the original set in the fixed frame, are then the input to the proposed method. The final result is a policy that takes as input the current positions relative to each frame and the current progress value φ, and outputs a desired change in the state $\Delta\boldsymbol{x}$.

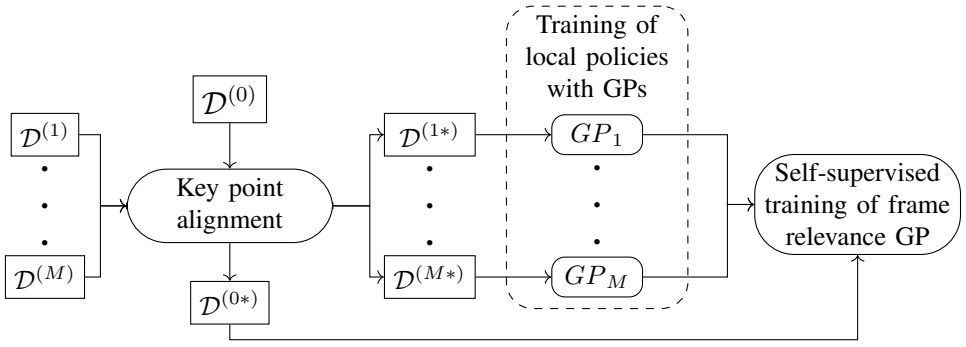


Fig. 2. TPGP pipeline, with the main sub-steps of the process indicated with rounded corner rectangles, and inputs and outputs indicated with regular rectangles.

B. Alignment of demonstrations

To successfully use the progress value in the state to learn the local policies, the demonstrations should first be aligned. An effective method that is commonly used for alignment is Dynamic Time Warping (DTW) [18]. However, in this case, trying to align all points between two demonstrations results in bad alignments, since the trajectories are not only dissimilar in timing but also in space.

Our assumption when performing the alignment is that in each of the local frames, the closest point in space must have happened at the same (normalized) time.

To find this alignment, for each ith demonstration in a set $D^{(m)}$, we find the index h of the closest point ${}^m\boldsymbol{x}_{i,\cdot}$ to every other element of other demonstrations ${}^m\boldsymbol{x}_{j,\cdot}$, i.e.,

$$ {}^m\boldsymbol{A}_{ij} = \arg\min_h \| {}^m\boldsymbol{x}_{i,h} - {}^m\boldsymbol{x}_{j,\cdot} \| \tag{3} $$

and the corresponding progress value at that index, i.e.,

$$ {}^m\boldsymbol{B}_{ij} = \varphi_{i,A_{ij}} \quad . \tag{4} $$

We can now define the keypoint progress value for demonstration i in frame m as the middle progress value of the closest points to any other demonstration, i.e.,

$$ \boldsymbol{P}_{im} = \text{median}\left({}^m\boldsymbol{B}_{i\cdot}\right). \tag{5} $$

The keypoints corresponding to each of these demonstrations and frames are visualized in the global frame in Fig. 4a. With a total of M frames, we will thus have M keypoints per demonstration.

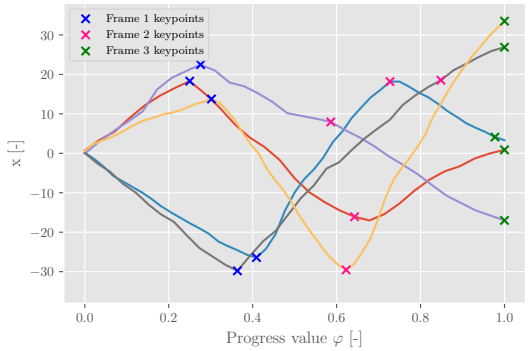
By resampling each demonstration, we enforce that the keypoints found in frame m are aligned to the same progress value, as shown in Fig. 4b. This results in the aligned demonstration sets $D^{(0*)}$, which can again be transformed to get $D^{(1*)}, \ldots, D^{(m*)}$.

C. Training the local policies

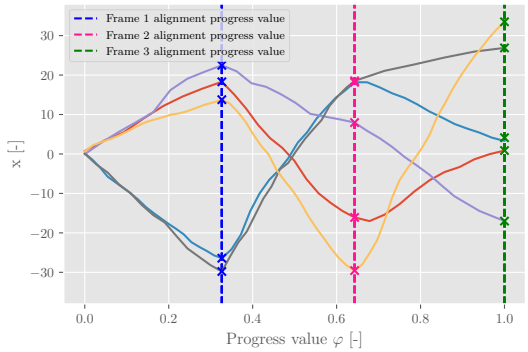
Local policies are encoded as a dynamical system,

$$ \Delta\boldsymbol{x} = f(\boldsymbol{x}). \tag{6} $$

The state $\boldsymbol{x}$ could have been represented using only position $\boldsymbol{\xi}$, or a combination of position and the progress value. Using only position could be advantageous since it means time misalignment of the demonstrations is not a problem.



(a) Before alignment.



(b) After alignment.

Fig. 3. Found keypoints for 5 example demonstrations and the resulting alignment. Only the x-coordinate is shown.

However, overlapping demonstrations in space can then introduce ambiguity in the training labels. Moreover, it would not be possible to teach picking skills where the robot needs to stop during the skill.

Thus, both Cartesian position $\boldsymbol{\xi}$ and a time-encoding variable were included in the state, i.e. $\boldsymbol{x} := [\boldsymbol{\xi}, \varphi]$. Specifically, a progress variable φ is used, which is the time normalized by the total time length of each demonstration, meaning $0 \leq \varphi \leq 1$.

The local policies are encoded using Gaussian Processes Regression [19], which allows us to have a quantification of uncertainties on the predictions, i.e.,

$$ f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}')) \tag{7} $$

where $k(\boldsymbol{x}, \boldsymbol{x}')$ is chosen to be a Matérn kernel and the mean

prior is the zero. When dragging the robot *far away* from labeled states, the agent will not update its time belief and will not move in any direction.

To find the posterior distribution we can use Bayes' theorem given the evidence of our data. However, since we are dealing with a (possibly) big dataset of demonstrations, the computation of the posterior becomes intractable. Hence, a variational approximation of the posterior distribution,

$$q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{S}), \tag{8}$$

is used, where $\boldsymbol{u}$ is a set of inducing variables that are located in $\boldsymbol{Z}$ and are distributed as a multivariate Gaussian with mean $\boldsymbol{m}$ and covariance $\boldsymbol{S}$. All the parameters of the variational distribution are fitted by maximizing the expected lower bound (ELBO) of the true log marginal likelihood of our label, see [20].

The predictive distribution $\boldsymbol{f}_*$ on a test point $\boldsymbol{X}_*$ is

$$p(\boldsymbol{f}_*) := \int p(\boldsymbol{f}_*|\boldsymbol{u})q(\boldsymbol{u})d\boldsymbol{u}. \tag{9}$$

Then, considering that $p(\boldsymbol{f}_*, \boldsymbol{u})$ is a joint multivariate normal[1], the result of the integral has a closed form, i.e.,

$$p(\boldsymbol{f}_*) = \mathcal{N}(\boldsymbol{A}\boldsymbol{m}, \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) + \boldsymbol{A}(\boldsymbol{S} - \boldsymbol{K}(\boldsymbol{Z}, \boldsymbol{Z}))\boldsymbol{A}^\top) \tag{10}$$

where $\boldsymbol{A} = \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{Z})\boldsymbol{K}(\boldsymbol{Z}, \boldsymbol{Z})^{-1}$. Given that we are modelling a multi-input multi-output dynamical system, every prediction returns the mean and variance of the desired state transition $p(\Delta\boldsymbol{x}_i) = \mathcal{N}(\mu_i, \sigma_i^2)$. The total variance is computed as $\sigma^2(\boldsymbol{x}) = \sum_i \sigma_i^2(\boldsymbol{x})$ given each of the i-th Cartesian output transitions.

If we consider the total uncertainty $\sigma(\boldsymbol{x}^2)$ as a quantification of the potential risk, we then seek to modify the autonomous dynamics to attract the system into regions of minimum risk. By adding the term

$$\Delta\boldsymbol{\xi}_i^{risk} = -\beta\sigma_i(\boldsymbol{x})\frac{\nabla_i\sigma^2(\boldsymbol{x})}{\|\nabla\sigma^2(\boldsymbol{x})\|}.$$

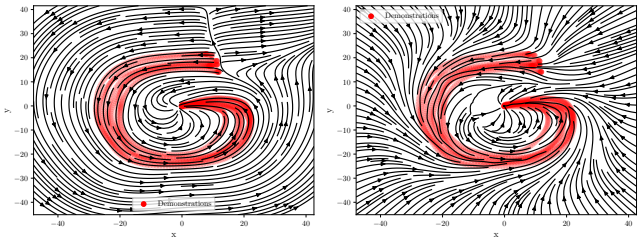
to each of the Cartesian transitions, we are *attracting* the autonomous system towards regions of minimum uncertainty with a step that is proportional to the standard deviation of each of the predictions. From an active inference perspective [21], this term tries to minimize the surprise associated with the belief and realizes action that would reject surprise proportionally to the surprise itself.

When dragging the dynamics outside the region of the demonstration, the standard deviation (std) prediction σ_i converges to the prior std, and for well-calibrated prior uncertainties, we know that

$$Pr(-2\sigma_i^{prior} < \Delta s_i(\boldsymbol{x}) < 2\sigma_i^{prior}) \simeq 0.96, \forall x \in \boldsymbol{X}$$

hence, by choosing β equal to 2, we are creating an attracting field that is calibrated, i.e. has similar energy to the learned dynamical systems and becomes predominant in regions where the dynamical system prediction converges to the prior. The authors of [5] and [22] showed how the variance minimization term is beneficial since it helps the robotic manipulator reject disturbances that might otherwise lead the robot to areas of high uncertainty, which could ultimately cause the robot to fail in performing its task due to out-of-distribution compounding errors.

[1]given the assumption that p(f, u) is a joint multivariate Gaussian distribution obtained from the Gaussian Process prior



(a) Without variance minimization (b) With variance minimization

Fig. 4. Streamplot regressed with a Variational Gaussian Process given the red demonstrations.

Fig. 4 illustrates the vector field of the learned dynamics with and without uncertainty minimization in reproducing the dynamics of drawing a letter "G" from the LASA dataset [3]. The Figure also highlights that the extra terms act as a stabilization term [4], [3], [23], by correcting learned diverging behaviours at the start of the demonstrations.

D. Self-Supervised Training of the Frame Relevance GP

Another Gaussian Process is regressed to determine the frame relevance as a function of the progress, φ,

$$\boldsymbol{\alpha} \sim \mathcal{GP}(0, k(\varphi, \varphi')). \tag{11}$$

However, as explained in the introduction, one of the goals of this method is to learn such a task without having to generate labels for the frame relevance in a supervised way. Given the local dynamics learned with respect to each of the frames, the frame relevance is regressed indirectly by using a new set of demonstrations that the local GPs have not been trained with. Each of the (trained) local GPs is then used to predict the local transition probability.

These predictions from the trained local GP policies are transformed to the global fixed frame using the transform ${}^m\boldsymbol{H}$ and weighted by each frame relevance. The likelihood of the relevance prediction is set as a softmax likelihood to ensure that the sum of weights predicted for each frame is not larger than one. The weighted sum of predicted transitions for each point in the demonstration is still Gaussian, i.e.,

$$p({}^0f_i) = \mathcal{N}({}^0\mu_i, {}^0\boldsymbol{\Sigma}_i) \tag{12}$$

where

$${}^0\mu_i = \sum_{m=1}^{M} {}^m\alpha_i \, {}^m\boldsymbol{H}^{-1} \, {}^m\mu_i({}^mx_i) \tag{13}$$

$${}^0\boldsymbol{\Sigma}_i = \sum_{m=1}^{M} {}^m\alpha_i \, {}^m\boldsymbol{H}^{-1} \, {}^m\boldsymbol{\Sigma}_i({}^mx_i) \, {}^m\boldsymbol{H}. \tag{14}$$

Given the recorded desired transition, ${}^0\Delta\boldsymbol{x}$, we can maximize the likelihood of each of the label transitions to belong to the predicted distribution $p({}^0f_i)$, i.e.

$$p({}^0\Delta\boldsymbol{x} \mid {}^0\boldsymbol{f}) = \prod_i^{n_d} p({}^0\Delta\boldsymbol{x}_i \mid {}^0f_i). \tag{15}$$

The likelihood maximization indirectly trains the variational distribution of the frame relevance GP such that the superposition of the local predicted distribution matches the distribution of the demonstrations. The optimization tries to find the parameters that would match the mean prediction while minimizing the total uncertainty of the prediction at each step, preferring the selection of the most confident (and correct!) local policy over the others.

IV. Experimental simulation results

During each demonstration, the agent's position and the initial position of any relevant coordinate frames are recorded. The simulation experiments are recorded through a 2D "drawing" interface where the demonstrations are given using a mouse. As a first performance metric we use the average of the minimum distances to each goal in the generated reproductions. The second performance metric is the average Fréchet distance [24] between the generated trajectories and the given demonstrations in the training case, and in the test cases we compare the generated trajectories to automatically generated expert trajectories.

When comparing the proposed model to other models, the Mann-Whitney U test is used to check whether the results from which the average metrics are calculated differ significantly. A threshold p-value of 0.05 is used, and metrics where the U test succeeded are highlighted in bold in the tables.

TPGP is trained on ten demonstrations for the two-frame task, where the origin of frame one is first approached, and then the origin of frame two is approached. Fig. 5 shows some example demonstrations for the same task, but with an additional third frame instead of only two.

The full model is tested on the ten training configurations and five test configurations for the two-frame task, both with and without variance minimization. Table I summarizes the results and highlights how the model with variance minimization performs better in almost every metric. Note also how, in the case where the model with variance minimization performed worse (average Fréchet distance for the test case), the difference in the results was not significant according to the Mann-Whitney U test.

A. Comparison with TPGMM

To evaluate the proposed method, its performance is compared to that of the state-of-the-art TPGMM algorithm [8], [25]. As mentioned in Section II, TPGMM fits GMMs on the data transformed to each of the relevant frames. Then at execution time, these can be transformed to the new configuration of the frames and multiplied together to find the new GMM for this configuration. Specifically, the resulting

TABLE I

Performance metrics for TPGP with and without variance minimization for the two frame task. Bold values indicate that the values used to calculate the average were significantly lower according to the Mann-Whitney U test.

	Average distance to goal 1 [-]		Average distance to goal 2 [-]		Average Fréchet distance [-]	
	train	test	train	test	train	test
TPGP w/o var. min.	0.64	0.68	3.03	7.36	5.93	10.08
TPGP with var. min.	**0.18**	**0.25**	**0.59**	**0.58**	**4.76**	14.12

TABLE II

Performance metrics for TPGP and TPGMM for the two frame task. Bold values indicate that the values used to calculate the average were significantly lower according to the Mann-Whitney U test.

	Average distance to goal 1 [-]		Average distance to goal 2 [-]		Average Fréchet distance [-]	
	train	test	train	test	train	test
TPGMM	**0.20**	0.27	0.17	0.14	5.51	10.64
TPGP (Ours)	0.38	0.29	**0.08**	**0.08**	**3.85**	**8.00**

TPGMM [26][2] is used to approximate a Hidden Markov Model (HMM). At execution time, the Viterbi algorithm is used to determine the most likely sequence of hidden states from a training demonstration, where each of these states corresponds to one of the Gaussian components of the GMM, and a linear-quadratic regulator (LQR) is employed to track the generated trajectories. Thus note that this version of TPGMM requires this additional sequence of states as input during execution, whereas TPGP only requires the new initial position of the frames.

Both TPGP and TPGMM are trained using 15 demonstrations, and then tested on 100 randomly generated test configurations. Table II shows the results for the two frame task, where for both the training and test frame configurations, TPGP performs better than TPGMM in four out of the six given metrics.

Similarly, for the three frame task, the results shown in Table III show that TPGP outperforms TPGMM in almost all the metrics. Fig. 5 shows example demonstrations along with the generated reproductions by TPGP and TPGMM, where the colors for the TPGP trajectory visually indicate the predicted frames' relevance. The relevance weights predicted by the frame relevance GP for this task are also visualized in Fig. 7, which indeed shows the desired switching behaviour in the correct order.

The top-left and bottom-left plots in Fig. 5 show how TPGMM sometimes deviates from the demonstrations, which

[2]An implementation can be found at `https://gitlab.idiap.ch/rli/pbdlib-python/-/blob/master/notebooks/pbdlib%20-%20Multiple%20coordinate%20systems.ipynb?ref_type=heads`

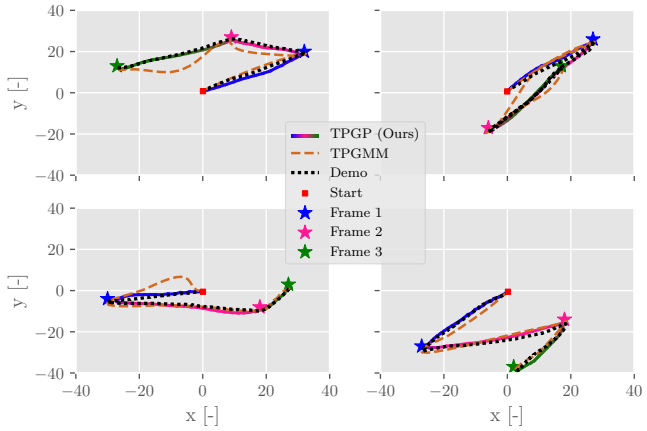


Fig. 5. Examples of training demonstrations for the three frame task, and the trajectories reproduced by TPGP and TPGMM for these configurations.

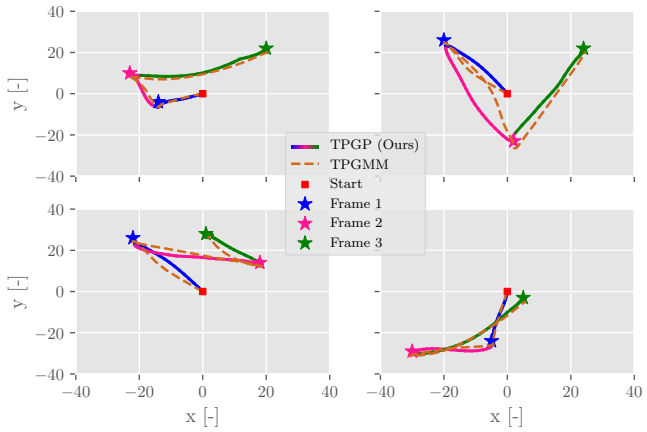


Fig. 6. Generated trajectories by TPGP and TPGMM for the three frame task for randomly generated configurations of the frames.

explains its worse Fréchet score. Fig. 6 also shows examples of generated trajectories for randomly generated test configurations.

V. Validation on a Robotic Manipulator

TPGP is also tested and validated using a 7-degree-of-freedom Franka-Emika manipulator. Cartesian impedance control is used to control the robot, where the end-effector is modeled as a spring-damper system. Kinesthetic demonstrations are provided by a user, and the recorded end-effector position is used as the input data.

At execution time, TPGP is used in an offline fashion: a trajectory is first generated and then executed by the controller as a sequence of attractors. For the re-shelving task, it is also necessary to generate values for the orientation and gripper commands of the end-effector. At each time step i, the most correlated point from the training data $\boldsymbol{X}$ to the current state $\boldsymbol{x}_i$ is found. The recorded gripper value and orientation at that most correlated training data point are then used for the execution at that time step.

TPGP is tested on a pick-and-place re-shelving task. A carton of milk must be picked up and then placed on a specific location on a shelf. Fiducial markers (AprilTags [27]) are used to localize the carton of milk and the placing goal during demonstrations and at execution time. An image of the setup for this task is shown in Fig. 8, where the base frame as well as the milk (frame 1) and placing goal (frame 2) are indicated on top of the fiducial markers, and Fig. 9 shows the full task through a sequence of images.

First, 10 demonstrations of this task are recorded while varying the position of both the object and the placing goal. Another 5 demonstrations are recorded to be used as a test set. Several TPGP models are then trained using four, six, and eight randomly chosen demonstrations out of the training demonstrations, and their performance is compared to a model trained with the complete dataset.

The results are shown in Table IV, where the same performance metrics explained in Section IV are used. Additionally, each of the models is used to try and complete the task for 10 new configurations, and the success rate is reported in the last column. As expected, the metrics improve as the number of demonstrations increases, especially the success rate.

To show the utility of the variance minimization in the

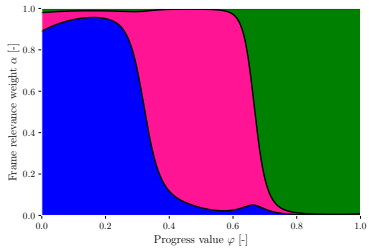


Fig. 7. Weights predicted by the Gaussian Process estimating the frame relevance for the three-frames task. Different colors refer to different frames.

TABLE III

Performance metrics for TPGP and TPGMM for the three frame task. Bold values indicate that the values used to calculate the average were significantly lower according to the Mann-Whitney U test.

	Average distance to goal 1 [-]		Average distance to goal 2 [-]		Average distance to goal 3 [-]		Average Fréchet distance [-]	
	train	test	train	test	train	test	train	test
TPGMM	0.82	0.68	0.52	0.5	0.44	0.35	5.00	12.44
TPGP (Ours)	0.70	0.71	**0.09**	**0.4**	**0.37**	**0.29**	4.13	**5.64**

TABLE IV

Performance metrics for TPGP trained with an increasing number of demonstrations.

	Average distance to goal 1 [cm]		Average distance to goal 2 [cm]		Average Fréchet distance [cm]		Task success rate [-]
	train	test	train	test	train	test	
4 Demos	1.0	1.2	2.4	3.1	10.2	20.5	50%
6 Demos	0.9	3.7	1.3	1.2	5.6	17.3	70%
8 Demos	0.5	3.8	1.2	2.5	8.3	10.6	80%
10 Demos	1.0	3.5	1.2	2.5	7.0	10.0	100%

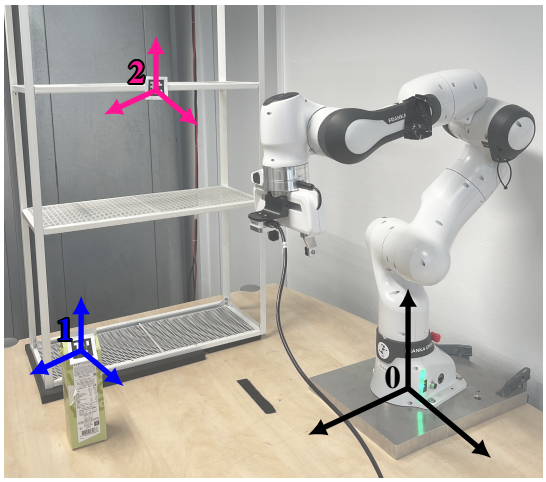


Fig. 8. Robotic setup for the re-shelving task, with visualizations of the base frame (0), the object frame (1), and the goal frame (2).

TABLE V

PERFORMANCE METRICS FOR TPGP WITHOUT VARIANCE MINIMIZATION TRAINED ON 4 AND 10 DEMONSTRATIONS.

	Average distance to goal 1 [cm]		Average distance to goal 2 [cm]		Average Fréchet distance [cm]		Task success rate [-]
	train	test	train	test	train	test	
4 Demos w/o var. min.	1.2	6.8	3.0	6.1	8.6	22.4	10%
10 Demos w/o var. min.	1.0	4.0	1.3	1.3	11.0	20.0	80%

local policies, we additionally test the performance of TPGP without this feature, trained on 4 and 10 demonstrations. Comparing these results shown in Table V with the results of the full model in Table IV, it is clear that the variance minimization improves the performance of the task. Note how it is especially helpful in the case with only 4 demonstrations, where task success rate improves from 10% to 50%.

Finally, we also present a comparison of the generated trajectories by TPGP and TPGMM for the robotic re-shelving task in Table VI. While they produce very similar results for the first metric, TPGP again scores better in the Fréchet distance. An example of the generated trajectories by each of the methods for a training configuration is also shown in Fig. 10. This Figure shows how TPGMM deviates from the demonstration when it approaches both frames, which explains its lower Fréchet distance scores.

VI. CONCLUSIONS AND FUTURE WORK

The presented method, TPGP, learns multi-reference frame skills directly from demonstration, without using task-specific heuristics or training labels on the frames' relevance. Local policies encode the dynamics relative to each frame, and a self-supervised approach is used to train the frame relevance GP which determines the frame relevance at each time step, to then select from the local policies. For both simulation and robotic re-shelving task experiments, TPGP is compared with the performance of another segmentation-

TABLE VI

PERFORMANCE METRICS FOR TPGP AND TPGMM FOR PICK AND PLACE TASK. BOLD VALUES INDICATE THAT THE VALUES USED TO CALCULATE THE AVERAGE WERE SIGNIFICANTLY LOWER ACCORDING TO THE MANN-WHITNEY U TEST.

	Average distance to goal 1 [cm]		Average distance to goal 2 [cm]		Average Fréchet distance [cm]	
	train	test	train	test	train	test
TPGMM	0.6	4.4	0.6	1.0	13.2	12.4
TPGP (Ours)	0.8	4.7	1.0	1.3	**8.9**	**9.7**

and heuristic-free model, TPGMM. In both cases, the TPGP model shows better performance. The distance to the task goals and the Frechét distance for demonstration reproductions are both smaller than for TPGMM.

A limitation inherent to TPGP (but also to TPGMM) is the need for diverse demonstrations. If several demonstrations are given but the configurations of the frames are very similar among these demonstrations, the model will fail to generalize to new configurations. A related issue is that a demonstrator might not know how to ensure the demonstrations and configurations are diverse. Thus a possible extension would be to integrate TPGP in an incremental learning framework, where additional training can easily be provided. An active learning element would go a step further, where the algorithm can request additional training data, potentially making use of the uncertainty quantification of Gaussian Processes.

REFERENCES

[1] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review ofControl, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 2020.

[2] C. Celemin, R. Pérez-Dattari, E. Chisari, G. Franzese, L. de Souza Rosa, R. Prakash, Z. Ajanović, M. Ferraz, A. Valada, and J. Kober, "Interactive imitation learning in robotics: A survey," *Foundations and Trends® in Robotics*, vol. 10, no. 1-2, pp. 1–197, 2022.

[3] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[4] R. Pérez-Dattari and J. Kober, "Stable motion primitives via imitation and contrastive learning," *IEEE Transactions on Robotics*, 2023.

[5] G. Franzese, A. Mészáros, L. Peternel, and J. Kober, "ILoSA: Interactive Learning of Stiffness and Attractors," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7778–7785.

[6] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023.

[7] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013, pp. 610–616.

[8] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent service robotics*, vol. 9, pp. 1–29, 2016.

[9] M. Wächter and T. Asfour, "Hierarchical segmentation of manipulation actions based on object relations and motion characteristics," in *2015 International Conference on Advanced Robotics (ICAR)*, pp. 549–556.

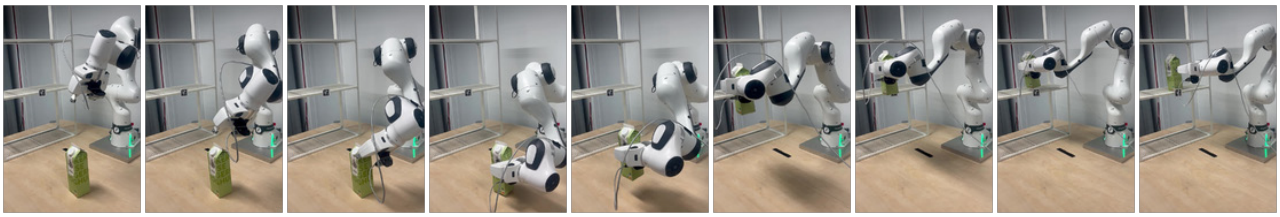


Fig. 9. Sequence showing the execution of the re-shelving task for a new, unseen configuration.

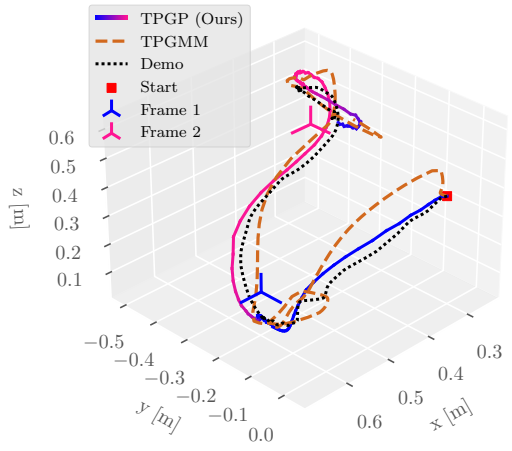


Fig. 10. Visualization of a provided demonstration for the re-shelving task, and the trajectories reproduced by TPGP and TPGMM for this training configuration. Note the deviation of TPGMM from the original demo close to the first frame.

[10] M. Mühlig, M. Gienger, and J. J. Steil, "Human-robot interaction for learning and adaptation of object movements," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4901–4907.

[11] B. Li, J. Li, T. Lu, Y. Cai, and S. Wang, "Hierarchical Learning from Demonstrations for Long-Horizon Tasks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4545–4551.

[12] J. Kober, M. Gienger, and J. Steil, "Learning movement primitives for force interaction tasks," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015, pp. 3192–3199, 06 2015.

[13] S. Manschitz, M. Gienger, J. Kober, and J. Peters, "Learning sequential force interaction skills," *Soft Robotics*, vol. 9, no. 2, 2020.

[14] L. Pais, K. Umezawa, Y. Nakamura, and A. Billard, "Learning robot skills through motion segmentation and constraints extraction," in *HRI Workshop on Collaborative Manipulation*. Citeseer, 2013, p. 5.

[15] A. L. P. Ureche, K. Umezawa, Y. Nakamura, and A. Billard, "Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1458–1471.

[16] S. Manschitz, M. Gienger, J. Kober, and J. Peters, "Mixture of attractors: A novel movement primitive representation for learning motor skills from demonstrations," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 926–933, 2018.

[17] J. Sun, J. Zhu, J. Kober, and M. Gienger, "Learning from few demonstrations with frame-weighted motion generation," in *18th International Symposium on Experimental Robotics (ISER)*, 2023.

[18] M. Müller, "Dynamic time warping," *Information Retrieval for Music and Motion*, pp. 69–84, 2007.

[19] C. Williams and C. Rasmussen, "Gaussian processes for machine learning, vol 2 Cambridge," *MA: MIT Press.*, 2006.

[20] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 351–360.

[21] P. Lanillos, C. Meo, C. Pezzato, A. A. Meera, M. Baioumy, W. Ohata, A. Tschantz, B. Millidge, M. Wisse, C. L. Buckley *et al.*, "Active inference in robotics and artificial agents: Survey and challenges," *arXiv preprint arXiv:2112.01871*, 2021.

[22] A. Mészáros, G. Franzese, and J. Kober, "Learning to Pick at Non-Zero-Velocity From Interactive Demonstrations," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6052–6059, 2022.

[23] N. Figueroa and A. Billard, "Locally active globally stable dynamical systems: Theory, learning, and experiments," *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 312–347, 2022.

[24] T. Eiter and H. Mannila, "Computing discrete Fréchet distance," Tech. Rep., 1994.

[25] S. Calinon, *Robot Learning with Task-Parameterized Generative Models*. Cham: Springer International Publishing, 2018, pp. 111–126. [Online]. Available: https://doi.org/10.1007/978-3-319-60916-4_7

[26] E. Pignat and S. Calinon, "Learning adaptive dressing assistance from human demonstration," *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.

[27] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.