IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Improved Generalization of Probabilistic Movement Primitives for Manipulation Trajectories

Xueyang Yao, Yinghan Chen, Bryan Tripp, *Member, IEEE*

***Abstract*—Imitation learning methods have proven effective in learning robotic tasks by leveraging multiple human-controlled demonstrations. However, existing approaches often struggle to generalize across a wide range of tasks, such as extrapolating to unseen object locations, incorporating via-point modulation, accurately modeling orientation, handling trajectories with multiple options, and capturing aiming actions. In this study, we propose a novel framework that combines ideas from task-parameterized Gaussian mixture models and probabilistic movement primitives to address these limitations and satisfy all the aforementioned properties within a single framework. We conduct comprehensive evaluations of our approach on four real-life tasks: pick-and-place, water pouring, shooting a hockey puck into a net, and sweeping.**

***Index Terms*—Programming by demonstration, imitation learning, few-shot learning, probabilistic movement primitives, task parameterized Gaussian mixture models.**

I. Introduction

PROGRAMMING by demonstration (PbD) [1] is an imitation learning approach that typically requires less than ten training examples per task. For example, [2] showed that a robot could learn a pick-and-place task on the basis of 4 to 7 examples. Similarly, in the work of [3], a collaborative object transportation task was learned using only 5 demonstrations. These approaches are examples of few-shot learning [4] that rely on engineering design to restrict the solution space. In contrast, reinforcement learning does not require human demonstrations, but it may require tens of thousands of trials, even for simple and narrow tasks such as bin picking.

In recent years, movement primitives (MPs) have garnered significant attention for their success in manipulation tasks. Dynamic Movement Primitives (DMPs) [5], [6] model a trajectory as a spring-damper dynamic system with a nonlinear forcing term. DMPs excel in generating smooth trajectories from any initial state. Probabilistic Movement primitives (ProMPs) [7], [8] directly learn a probabilistic model over the trajectories. ProMPs not only provide an estimate of the underlying trajectory derived from multiple demonstrations but also capture its variability through the covariance matrix. Probabilistic Dynamic Movement Primitives (ProDMPs) [9] provide a unified framework that combines the principles of both DMPs and ProMPs. ProDMPs achieve this by recovering a linear basis-function representation for trajectories through the solution of a dynamical system. Notably, DMPs, ProMPs, and ProDMPs all share the requirement for manually defining basis functions. Kernelized Movement Primitives (KMP) [3] and Gaussian Process (GP) based methods [10], [11] alleviate the need for manual basis function definition by employing kernel techniques. It is worth noting that all of the aforementioned methods achieve adaptation by relying on via-points, a concept that may not align naturally with real-life manipulation tasks. Determining the appropriate via-points typically involves expert knowledge and heuristics.

Task-parameterized Gaussian Mixture Models (TP-GMM) [12] employ a distinctive approach for adaptation. TP-GMM leverages multiple reference frames to represent the end-effector's trajectory. Within each local reference frame, a statistical model is learned, and the global trajectory model is obtained by multiplicatively combining the local models. This adaptation method aligns more naturally with manipulation tasks since it allows TP-GMM to adapt based on the poses of objects. These poses can be acquired using a vision module or by attaching markers to objects, eliminating the need for additional heuristics. Furthermore, TP-GMM shows robust extrapolation capabilities by encoding movements from multiple reference frames, enabling it to adapt to object poses beyond the training distribution. However, it is important to note the time-based TP-GMM approach is unsuitable for cyclical movements and struggles to handle tasks with multiple modes, such as multiple paths around an obstacle. In contrast, dynamic-based TP-GMM models the joint distribution of dynamic features, including position, velocity, and acceleration. This enables the dynamic-based TP-GMM to effectively handle cyclical movements and trajectories with multiple options. [13] extends the TP-GMM approach by modeling quaternions on a 3-dimensional Riemannian manifold that respects the geometry of quaternion space. However, computing Gaussian product and conditioning on a Riemannian manifold requires iterative computations on the tangent space.

Inspired by TP-GMM, Local Kernelized Movement Primitives (KMPs) [3] were introduced to enhance KMP's extrapolation capabilities. Furthermore, KMPs respecting the geometry of quaternions were introduced in [14]. However, akin to time-based TP-GMM, KMPs are unable to infer and model multiple modes from the dataset.

Another family of approaches is to use neural networks (NNs) to abstract the task parameters from high-dimensional modalities, e.g. scene images, to support end-to-end trajectory

Manuscript received: June 30, 2023; Revised: September 27, 2023; Accepted: October 29, 2023.

This paper was recommended for publication by Editor Aleksandra Faust upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the Natural Sciences and Engineering Research Council of Canada and Applied Brain Research.

The authors are with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: x35yao@uwaterloo.ca, y977chen@uwaterloo.ca, bptripp@uwaterloo.ca).

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

learning. These approaches [15]–[18] eliminate the need for manually engineered task parameters. However, the task of inferring parameters from a restricted dataset is inherently challenging, and modeling trajectories atop it presents an even greater difficulty. In [19], representation learning and behavior learning are decoupled. When presented with a new scene image, this approach identifies the nearest neighbors in the representation space, and the agent's behavior is obtained by a weighted average of actions from these neighbors. However, interpolating between nearest neighbors poses challenges when tested outside the training distribution [20].

Our proposed method, Task-parameterized Probabilistic Movement Primitives (TP-ProMP), addresses limitations of previous approaches by introducing multiple local reference frames to the ProMP framework. By combining the strengths of ProMP and TP-GMM, TP-ProMP offers several key advantages, including extrapolation capability, adaption to different task parameters, and via-point modulation. Furthermore, TP-ProMP demonstrates accurate learning of 6D end effector trajectories. To enhance the modeling of time-based trajectories with multiple modes, we extend TP-ProMP by representing weights as a mixture of Gaussian distributions. While previous works have explored this idea [21], our method introduces a more robust approach. In addition, we incorporate additional reference frames that encompass pairs of objects. We show that this extension yields more accurate trajectory predictions in aiming tasks. Finally, departing from recent practice in PbD, we do not affix markers to objects but instead sense their configurations using deep networks, eliminating a physical task constraint and broadening the range of tasks our method can handle. This approach introduces vision-related errors. However, these additional errors are consistent across different PbD methods, allowing a fair comparison. Overall, the results we report here are practical lower bounds on the performance of these methods.

We make our demonstration datasets public to facilitate future work (https://github.com/x35yao/TP-ProMP/tree/main/data).

II. Methods

A. Tasks

We evaluated PbD methods with 4 different tasks:

- Pick-and-place task testing extrapolation performance.
- Water pouring task consisting of different modes of pouring from both the left and right of the cup. This task evaluated the methods' ability to accurately model orientation and its resilience to mode collapse.
- Shooting hockey puck task involves modeling the motion of the gripper holding a hockey stick and striking the puck into a net from various directions.
- Sweeping task involving modeling the motion of the gripper holding broom and cleaning rubbish into a dustpan.

Figure 1 shows images of these tasks.

B. Data collection

The workspace was instrumented with both a Zed2 stereo camera (used to sense object positions and orientations) and an NDI Polaris optical motion tracker (used to sense gripper position and orientation). Two multi-marker "tools" were mounted to the gripper handle at different positions to avoid occlusion such that the motion tracker could track the 6DOF configuration of the gripper [22]. The camera and motion tracker were synchronized and both signals were resampled to a common time step of 0.08 seconds. DeepLabCut [23] was employed to estimate the two-dimensional positions of object parts in the image space of both the left and right cameras. We derived their three-dimensional coordinates by triangulating these positions.

We used the following procedure to robustly estimate object poses even when certain parts were occluded. Prior to each task, the relevant objects were placed on the table with all object parts visible. The triangulated 3D object-part positions, denoted as $\boldsymbol{p_i}_{i=1}^{N}$, were used to create an object template. The object center was determined as the mean of all object-part positions, and a coordinate system was assigned to each object template. The pose of each object template in the global reference frame was represented by a rotation matrix $\boldsymbol{A_1}$ and a translation vector $\boldsymbol{b_1}$. By aligning the template's axes with the global reference frame, $\boldsymbol{A_1}$ was simplified as an identity matrix and $\boldsymbol{b_1}$ represented the center position.

During each demonstration, a different set of object-part positions $\boldsymbol{p'_i}_{i=1}^{N'}$ was obtained, with $N' \leq N$. We then employed a rigid 3D transformation algorithm [24] to approximate the rotation matrix $\boldsymbol{A_2}$ and translation vector $\boldsymbol{b_2}$ that minimized the least squares error:

$$err = \Sigma_{i=1}^{N'} ||\boldsymbol{p'_i} - (\boldsymbol{A_2 p_i} - \boldsymbol{b_2})||^2 \tag{1}$$

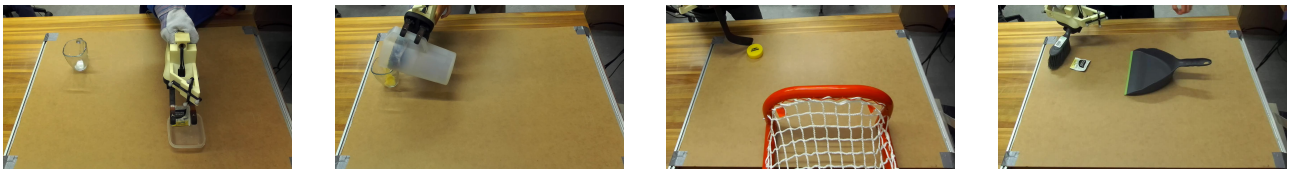
$\boldsymbol{A_2}$ and $\boldsymbol{b_2}$ can also be interpreted as representing the object's pose in the object template's reference frame. We obtained the object's pose (task parameters) in the global reference frame as $\boldsymbol{A} = \boldsymbol{A_1 A_2}$ and $\boldsymbol{b} = \boldsymbol{A_1 b_2} + \boldsymbol{b_1}$. These task parameters were used to convert trajectories between local and global reference frames as described in [12]. For tasks where the object's orientation did not significantly impact the trajectory modeling, $\boldsymbol{A}$ was simplified as an identity matrix.

C. Preprocessing

Dynamic time warping was used to realign the trajectories and normalize the durations of different demonstrations of the same task. The demonstration with the median duration was used as the reference demonstration that others were realigned around. The minimized cost metric was the gripper's speed difference versus that in the reference demonstration. To avoid extreme jumps in trajectory, step sizes were constrained within 1/2 and 2 times corresponding step sizes in the reference trajectory [25].

To deal with outlier demonstrations from noisy sensors, outliers were removed based on whether the corresponding trajectory's start and end points deviated significantly from the mean of the current set of demonstrations by standard deviation threshold of 3. Due to the more extreme outliers skewing the distributions, this was repeated after the detected outliers were removed and halted when the next iterations generated no new outliers.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.



(a) Pick and place (b) Pour water (c) Shoot Puck (d) Sweep

Fig. 1. Demonstration of the tasks. (a) The pick-and-place task involves placing the gripper above a teabag, grasping and moving the gripper above a cup, and placing the teabag into the cup. (b) The water-pouring task involves picking up a pitcher filled with water, pouring water into a cup, and then placing the pitcher back down on a table. (c) The shooting-puck task where the demonstrator shoots a puck into a net on the table. (d) The sweeping task where the demonstrator cleans a rubbish into a dustpan.

D. Task-parameterized Probabilistic Movement Primitive (TP-ProMP)

1) Learning from Demonstrations:

Our methods are based on both Probabilistic Movement Primitives (ProMPs) and TP-GMMs. A trajectory $\tau = \{\boldsymbol{x}_t\}_{t=1}^{T}$ represents a trajectory in the global reference frame at different moments in time, where $\boldsymbol{x}_t$ is a D dimensional vector. In other words, a trajectory is represented as the matrix $\tau \in \mathbb{R}^{D\times T}$. In this paper, if both the position and orientation are modeled, $\boldsymbol{x}_t$ is a 7D dimensional vector, where the first three dimensions represent the position of the end-effector and the remaining four dimensions represent the quaternions that describe its orientation. When only the end-effector's position is modeled, $\boldsymbol{x}_t$ is simplified to a three-dimensional vector. We assigned R local reference frames to objects in the scene that are related to the task. For the jth reference frame, we obtained the task parameters $\boldsymbol{A_j}$ and $\boldsymbol{b_j}$ following the procedure outlined in II-B. By utilizing these task parameters, we converted the trajectory from the global reference frame to the jth local object reference frame, resulting in $\boldsymbol{x}_t^{(j)}$, as explained in [12]. Here, the superscript j signifies that the trajectory is expressed in the jth reference frame.

Trajectories from all reference frames were then stacked to get $\tau' = \{\boldsymbol{\chi}_t\}_{t=1}^{T}$, where $\boldsymbol{\chi}_t = \begin{bmatrix} x_{1,t}^1 & \dots & x_{D,t}^1 & \dots & x_{1,t}^R & \dots & x_{D,t}^R \end{bmatrix}^\top$ combines trajectories from all reference frames and $x_{d,t}^j$ denotes the gripper trajectory value for the dth dimension in the jth reference frame. $\tau' \in \mathbb{R}^{DR\times T}$ can be interpreted as a trajectory with $M = DR$ dimensions. We then fit τ' to a ProMP model. As described in [7], ProMP represents a trajectory with a weight vector and variability of the given demonstrations is modelled in the weight space. The combined weight vector $\boldsymbol{\omega} = \begin{bmatrix}\boldsymbol{\omega}_1^\top, & \boldsymbol{\omega}_2^\top, & \dots & ,\boldsymbol{\omega}_M^\top\end{bmatrix}^\top$ for a trajectory consists of M weight vectors $\boldsymbol{\omega}_m$ that correspond to each dimension m. The trajectory at time t given the weight vector $\boldsymbol{\omega}$ is defined as

$$\boldsymbol{\chi}_t = \boldsymbol{\Phi}_t\boldsymbol{\omega} + \boldsymbol{\epsilon}_\chi, \tag{2}$$

where, $\boldsymbol{\epsilon}_\chi \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$ is Gaussian white noise, $\boldsymbol{\Phi}_t$ is an $M \times KM$ matrix defined as

$$\boldsymbol{\Phi}_t = \begin{bmatrix} \phi_1(t) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \phi_M(t) \end{bmatrix}, \tag{3}$$

$\phi_m(t)$ defines the K dimensional time-dependent basis vector for the mth dimension and K defines the number of basis functions. In this work, we simplified $\boldsymbol{\Sigma}_\epsilon$ to be a diagonal matrix. For basis functions, we used 21 evenly spaced Gaussian radial functions and polynomial terms up to 3rd-order. The Gaussian widths were set empirically task-by-task. The probability of observing τ' is defined as

$$p(\tau' \mid \boldsymbol{\omega}) = \prod_t \mathcal{N}(\boldsymbol{\chi}_t \mid \boldsymbol{\Phi}_t\boldsymbol{\omega}, \boldsymbol{\Sigma}_\epsilon) \tag{4}$$

Now, suppose we have N demonstrations and use subscript $n \in \{1 \dots N\}$ to identify each demonstration. For the nth demonstration, $\boldsymbol{\chi}_{nt}$ represents the combined trajectory state from all reference frames at time t, and $\boldsymbol{\omega}_n$ is the compact representation of the combined trajectory. Assume $\boldsymbol{\omega}_n$ is sampled from a Gaussian distribution $\mathcal{N}(\boldsymbol{\omega}_n \mid \boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega)$, the likelihood of observing N combined trajectories is given by

$$p(\boldsymbol{X} \mid \boldsymbol{\theta}_\omega) = \prod_{n=1}^{N} \int \mathcal{N}(\boldsymbol{\omega}_n \mid \boldsymbol{\theta}_\omega) \prod_{t=1}^{T_n} \mathcal{N}(\boldsymbol{\chi}_{nt} \mid \boldsymbol{\Phi}_t\boldsymbol{\omega}_n, \boldsymbol{\Sigma}_\epsilon) d\boldsymbol{\omega}_n, \tag{5}$$

where $\boldsymbol{\theta}_\omega = \{\boldsymbol{\mu}_\omega, \boldsymbol{\Sigma}_\omega\}$, $\boldsymbol{X}$ is a set of N combined trajectories $\boldsymbol{\chi}_n$, and T_n denotes the trajectory length of the nth demonstration. The parameters $\boldsymbol{\mu}_\omega$, $\boldsymbol{\Sigma}_\omega$, and $\boldsymbol{\Sigma}_\epsilon$ can be learned through the Maximum A-Posteriori estimates(MAP) algorithm [26].

Given the learned parameters, the combined trajectory at any time point t can be evaluated by

$$p(\boldsymbol{\chi}_t \mid \boldsymbol{\theta}_\omega) = \int \mathcal{N}(\boldsymbol{\omega} \mid \boldsymbol{\theta}_\omega)\mathcal{N}(\boldsymbol{\chi}_t \mid \boldsymbol{\Phi}_t\boldsymbol{\omega}, \boldsymbol{\Sigma}_\epsilon) d\boldsymbol{\omega} \tag{6}$$

$$= \mathcal{N}(\boldsymbol{\chi}_t \mid \boldsymbol{\Phi}_t^\top\boldsymbol{\mu}_\omega, \boldsymbol{\Phi}_t^\top\boldsymbol{\Sigma}_\omega\boldsymbol{\Phi}_t + \boldsymbol{\Sigma}_\epsilon). \tag{7}$$

The distribution of $\boldsymbol{\chi}_t$ is a Gaussian distribution with mean $\boldsymbol{\mu}_{\chi t} = \boldsymbol{\Phi}_t^\top\boldsymbol{\mu}_\omega$ and covariance $\boldsymbol{\Sigma}_{\chi t} = \boldsymbol{\Phi}_t^\top\boldsymbol{\Sigma}_\omega\boldsymbol{\Phi}_t + \boldsymbol{\Sigma}_\epsilon$. $\boldsymbol{\mu}_{\chi t}$. $\boldsymbol{\Sigma}_{\chi t}$ can be decomposed by corresponding reference frames $\boldsymbol{\mu}_{\chi t} = \begin{bmatrix}\boldsymbol{\mu}_{xt}^{(1)}, & \dots & ,\boldsymbol{\mu}_{xt}^{(R)}\end{bmatrix}$ and $\boldsymbol{\Sigma}_{\chi t} = \begin{bmatrix} \boldsymbol{\Sigma}_{xt}^{(1)} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \boldsymbol{\Sigma}_{xt}^{(R)} \end{bmatrix}$. At time t, we obtain a Gaussian distribution with mean $\boldsymbol{\mu}_{xt}^{(j)}$, and covariance matrix $\boldsymbol{\Sigma}_{xt}^{(j)}$ for the trajectory in the jth reference frame. $\boldsymbol{\mu_{xt}}$. We then predict the trajectory in the global reference frame. This involves converting the local models to the global reference frame and multiplicatively combining the resulting Gaussian models. Therefore, given new task parameters $A^{(j)}$ and $b^{(j)}$ for the jth reference

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

frame, we first convert $\boldsymbol{\mu}_{xt}^{(j)}$ and $\boldsymbol{\Sigma}_{xt}^{(j)}$ to the global reference frame as $\boldsymbol{\mu}_{xt,j}$ and $\boldsymbol{\Sigma}_{xt,j}$ respectively, where the subscript j indicates it is converted from the jth reference frame. The predicted covariance matrix in the global reference frame can be computed with

$$\boldsymbol{\Sigma}_{xt} = (\sum_{j=1}^{R} \boldsymbol{\Sigma}_{xt,j}^{-1})^{-1}, \tag{8}$$

and the predicted mean in the global reference frame can be computed with

$$\boldsymbol{\mu}_{xt} = \boldsymbol{\Sigma}_{x,t} \sum_{j=1}^{R} \boldsymbol{\Sigma}_{xt,j}^{-1} \boldsymbol{\mu}_{xt,j} \tag{9}$$

2) Via-point modulation:

As previously discussed, ProMP achieves adaptation by implicitly conditioning on a via-point at a specific time t. While this approach may not be natural for adapting to new object poses, there are situations where implicit conditioning can be beneficial. For instance, when a new obstacle is introduced into the environment after the model has been trained, conditioning the model on a position far from the obstacle allows it to continue performing the task without requiring the learning of a new model. Similar to [7], the proposed method achieves via-point modulation by conditioning. Given a desired point $\boldsymbol{x}_t^*$ in the global reference frame, we can compute the combined desired point in all reference frames $\boldsymbol{\chi}_t^*$. Then, for a specific accuracy matrix of the desired observation $\boldsymbol{\Sigma}_{\chi}^*$, the conditional distribution $p(\boldsymbol{\omega} \mid \boldsymbol{\chi}_t^*)$ is Gaussian with mean and variance [7]

$$\boldsymbol{\mu}_{\omega}^{[new]} = \boldsymbol{\mu}_{\omega} + \boldsymbol{\Sigma}_{\omega} \boldsymbol{\Phi}_t (\boldsymbol{\Sigma}_{\chi}^* + \boldsymbol{\Phi}_t^{\top} \boldsymbol{\Sigma}_{\omega} \boldsymbol{\Phi}_t)^{-1} (\boldsymbol{\chi}_t^* - \boldsymbol{\Phi}_t^{\top} \boldsymbol{\mu}_{\omega}) \tag{10}$$

$$\boldsymbol{\Sigma}_{\omega}^{[new]} = \boldsymbol{\Sigma}_{\omega} - \boldsymbol{\Sigma}_{\omega} \boldsymbol{\Phi}_t (\boldsymbol{\Sigma}_{\chi}^* + \boldsymbol{\Phi}_t^{\top} \boldsymbol{\Sigma}_{\omega} \boldsymbol{\Phi}_t)^{-1} \boldsymbol{\Phi}_t^{\top} \boldsymbol{\Sigma}_{\omega}. \tag{11}$$

Figure 2 provides a visual example of conditioning a TP-ProMP model to different via points. TP-ProMP maintains local low-variance movement patterns while passing through the via points. This demonstrates the ability of TP-ProMP to adapt and incorporate via points with minimal deviations from the learned movement pattern.

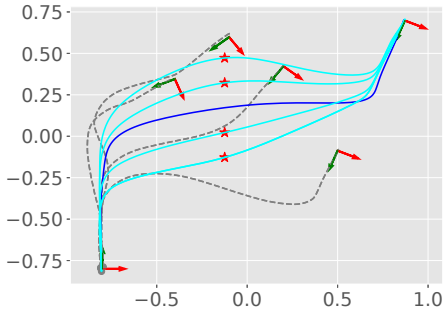


Fig. 2. Conditioning on different via-points. The dashed gray trajectories are the training demonstrations. The blue trajectory is the prediction of the original model given new task parameters. The cyan trajectories are modulated trajectories conditioned on different via points (red stars).

E. Extensions of TP-ProMP

1) Model $\boldsymbol{\omega}$ as a mixture of Gaussians:

To accommodate tasks with multiple modes, we extended our method by representing the weight vector $\boldsymbol{\omega}_n$ as a mixture of Gaussians. In previous approaches, such as the one proposed in [21], a point estimate of $\boldsymbol{\omega}_n$ was first obtained using least squares, and then an Expectation-Maximization (EM) algorithm was used to learn a Gaussian Mixture Model (GMM) in the weight space. However, this point estimate ignored the uncertainty of $\boldsymbol{\omega}_n$ and resulted in reduced robustness in tasks with high sensor noise. Additionally, the Maximum Likelihood Estimate (MLE) used in the M-step of the EM algorithm led to numerical issues when computing the covariance matrix $\boldsymbol{\Sigma}_{\omega}$.

To overcome these limitations, an alternative approach was presented in [26], which incorporated uncertainty estimation by iteratively updating $\boldsymbol{\omega}_n$ in the E-step based on the optimized parameters from the previous M-step. Furthermore, a prior distribution was introduced to achieve a Maximum A Posteriori (MAP) estimate and address the numerical issues. However, the method in [26] still modeled $\boldsymbol{\omega}$ as a Gaussian distribution, making it incompatible with trajectories exhibiting multiple modes.

In our extension, we have made modifications to the algorithm proposed in [26] by introducing a mixture of Gaussians model for $\boldsymbol{\omega}$. The extended algorithm consists of two steps. In the first step, we employed an EM algorithm to cluster the N demonstrations into C groups. During the E-step, each Gaussian component yields an estimated $\overline{\boldsymbol{\omega}}_{ni}$ as described in [26]. The subscript i denotes it is for the ith component. The overall estimated value $\overline{\boldsymbol{\omega}}_n$ was then computed as the weighted sum of $\overline{\boldsymbol{\omega}}_{ni}$ across all Gaussian components. In the M-step, we used a Maximum A Posteriori (MAP) estimate to update the parameters for each Gaussian component. We utilized a Normal-Inverse-Wishart distribution with parameters k_0, $\boldsymbol{m}_0$, v_0, and $\boldsymbol{S}_0$. In this work, as guided by the recommendations in [26], we set $k_0 = 0$, $\boldsymbol{m}_0 = \boldsymbol{0}$, $v_0 = \dim(\boldsymbol{\omega}) + 1$, and $\boldsymbol{S}_0 = (v_0 + KM + 1)\text{blockdiag}(\boldsymbol{\Sigma}_{\omega i}^*)$, where $\boldsymbol{\Sigma}_{\omega i}^*$ is the MLE of the covariance matrix of the ith Gaussian component. In this step, since our goal was simply to cluster demonstrations based on execution modes, we forced the covariance matrix of each Gaussian component $\boldsymbol{\Sigma}_{\omega i}$ to be diagonal for better cluster performance.

In the second step, C ProMP models were further trained on the corresponding data using the parameters $\boldsymbol{\mu}_{\omega}$, $\boldsymbol{\Sigma}_{\omega}$, and $\boldsymbol{\Sigma}_{\epsilon}$ obtained from the first step as initial parameters, following the training procedure described in [26].

2) Paired-object parameterized ProMP:

We extended the TP-ProMP model to allow greater flexibility which we will now describe. For each source object $\boldsymbol{o} \in \boldsymbol{O}$, where $\boldsymbol{O} = \{\boldsymbol{o_i}\}_{i=1}^{R}$ and some target object $\boldsymbol{o'} \in \boldsymbol{O} \setminus \{\boldsymbol{o}\}$, we generated a reference frame $(\boldsymbol{o}, \boldsymbol{o'})$. The origin of the new coordinate system in the pairwise reference frame was the center of the source object $\boldsymbol{o}$. The axes were aligned as follows. The x-axis was in the direction of the line passing from the center of $\boldsymbol{o}$ to that of $\boldsymbol{o'}$. The new y-axis was in the direction of a unit vector $\boldsymbol{u'}$ orthogonal to the x-axis such that $\max_{\boldsymbol{u'}=\boldsymbol{u}} \boldsymbol{u} \cdot \boldsymbol{u}_{\boldsymbol{height}}^{\boldsymbol{world}}$, where $\boldsymbol{u}_{\boldsymbol{height}}^{\boldsymbol{world}}$ is the unit vector in the world coordinate parallel to the height axis. The z-axis

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

was given by the right-hand rule. This procedure produced $R_{pop} = R^2$ reference frames, consisting of R individual reference frames centered on the object and aligned with world axes, plus $R(R-1)$ pairwise reference frames. To counter the quadratic growth of complexity and eliminate useless reference frames, we introduced a selection process to pick out the most task-relevant frames for each object. The mean and variance of the demonstration trajectories in each candidate reference frame were calculated, and the candidate for a given object with the tightest clustering at any moment in the trajectory (i.e. the smallest variance over candidates and time) was selected. The reference frames selected by this method may align with a task constraint at some point of time during the task. The resulting model has the same number of reference frames and parameters as the TP-ProMP model.

F. Evaluation

To evaluate the effectiveness of our proposed method for a given task, we randomly selected N demonstrations from our dataset as the training set, 1 demonstration as the validation trajectory, and 1 demonstration as the test trajectory. The validation and test trajectories were not in the training set. As the task objects were positioned differently in each demonstration, the validation and test demonstrations did not resemble the training set. To assess the accuracy of the prediction, the Euclidean distance was used as a similarity measure for position. As for orientation, the norm of the difference of the quaternions was utilized as the metric [27]. We report the mean of this metric over the trajectory as it is a simple and general reflection of the performance across the trajectory as a whole. Additionally, we report the similarity measure at key points such as the start, end, and middle. We repeated this process 100 times for all tasks and computed the average and standard deviation. We compared our proposed method with the original ProMP model, the time-based TP-GMM model, the dynamic-based TP-GMM model, and the KMP model. When comparing with the KMP and TP-GMM models (time-based and dynamic-based), we trained several models with the number of components ranging from 2 to 10. The model with the best position prediction was chosen for comparison.

Our basic TP-ProMP model models the 3D gripper position only, with $\boldsymbol{\omega}$ modeled as a Gaussian distribution and only object reference frames used. This basic model was used in the pick-and-place task. Depending on the specific task demands, we extended this basic TP-ProMP model in several ways:

1) 7D Trajectories: Including 3D positions and 4D Quaternions (Pouring task).
2) Multi-Modal Trajectories: Modeling $\boldsymbol{\omega}$ as GMM to handle multi-modal trajectories effectively (Pouring task).
3) Paired-object parameterization: This was used for the shooting and sweeping tasks.

III. RESULTS

A. Pick and place

During data collection, the table was divided into two areas, and demonstrations with both the teabag and cup on the right half were labeled as the training data pool, while those with at least one object on the left half were labeled as the test data pool. This separation ensured that a portion or the entirety of a trajectory from the test pool falls outside the distribution of the training pool. We then assigned local reference frames to the cup and the teabag to train a TP-ProMP model.

The evaluation was conducted as described in Section II-F. To adapt to new objects' positions, we conditioned the ProMP model as described in [7]. Specifically, we conditioned the starting point of the trajectory to be 20 mm above the teabag and the endpoint of the trajectory to be 20 mm above the cup. All 5 models used in the experiment were trained on 6 demonstrations. However, the ProMP model was trained only on trajectories in the global reference frame. On the other hand, TP-ProMP, TP-GMM, and KMP models were trained on trajectories in the cup and teabag's reference frames.

Figure 3a demonstrates the performance comparison between the 5 models for one specific test case. The figure shows that the TP-ProMP, KMP, and TP-GMM models outperform the ProMP model. The trajectories predicted by TP-ProMP, KMP, and TP-GMM models are closer to the ground truth in this extrapolation test, particularly at the beginning and end of the trajectory. This observation is significant in this task, as accurately predicting the start and end of the trajectory is crucial to ensure the robot successfully grasps the teabag and places it into the cup. The average performance over 100 tests is reported in Table I. As seen in Table I, TP-ProMP outperforms other models in general, except at the start of the trajectory where TP-ProMP shows slightly worse results compared to the KMP and the TP-GMM models.

ProMP's performance suffers due to two primary reasons. Firstly, it struggles when the conditioned point lies beyond the training distribution. Additionally, it heavily relies on heuristics, such as the gripper's assumed 20 mm elevation above objects at the trajectory's start and end points. These heuristic dependencies hinder ProMP's adaptability and limit its application, particularly when precise timing at via points is uncertain, as seen in the water pouring task.

B. Water pouring

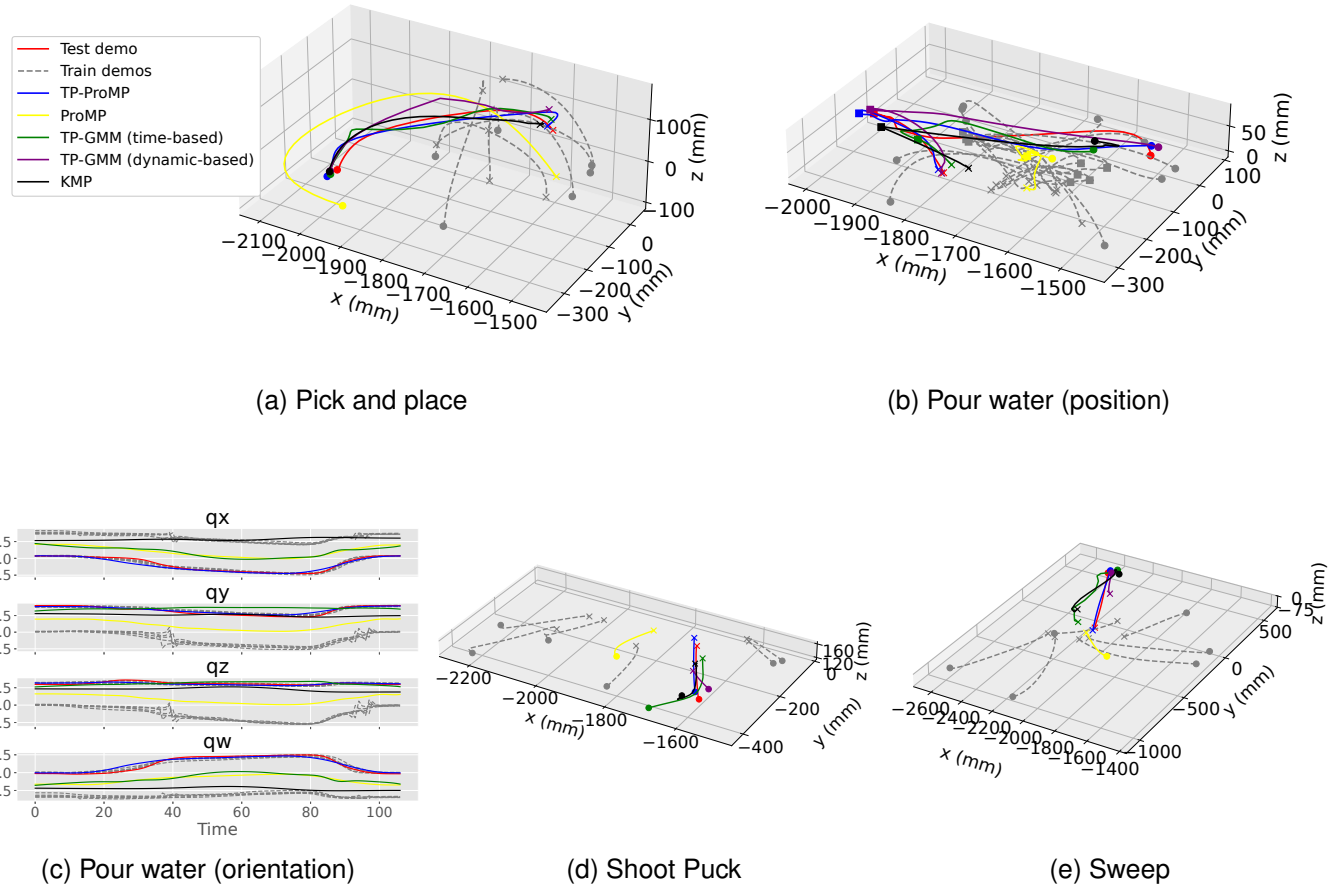
Demonstrations for this task used millet for safety, with executions in both left and right hands, resulting in two modes. Compared to pick-and-place, the pouring task is more challenging due to: 1) Gripper orientation changes significantly, requiring a 7D trajectory representation. 2) Involvement of two pouring modes, necessitating multi-modal trajectory modeling for accurate capture.

For position prediction, we compared the performance of TP-ProMP with ProMP, TP-GMM (time-based), and TP-GMM (dynamic-based), and KMP models. All models were trained on a total of 10 demonstrations with equal numbers of left- and right-handed executions. The ProMP model was not adapted, meaning it only captured the mean behavior of the training demonstrations. As discussed in Section III-A, ProMP suffers from the limitation that adapting to new object positions is achieved through implicit conditioning on position and time. In this particular task, this implies conditioning

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE I
MEAN +/- STANDARD DEVIATION DISTANCE FROM GROUND TRUTH FOR PROMP, TP-GMM, KMP, AND TP-PROMP FOR THE PICK-AND-PLACE TASK.

	Start (mm)	End (mm)	Average (mm)
ProMP	107.78 ± 7.99	137.84 ± 18.10	250.68 ± 137.13
TP-GMM (time-based)	28.22 ± 14.90	74.11 ± 48.06	58.76 ± 21.43
TP-GMM (dynamic)	27.54 ± 15.56	129.42 ± 100.18	83.45 ± 43.01
KMP	$\mathbf{25.80 \pm 11.39}$	73.99 ± 36.44	57.48 ± 21.62
TP-ProMP	32.62 ± 8.54	$\mathbf{48.60 \pm 20.68}$	$\mathbf{48.51 \pm 12.21}$



(a) Pick and place

(b) Pour water (position)

(c) Pour water (orientation)

(d) Shoot Puck

(e) Sweep

Fig. 3. Trajectory predictions of the 5 models for 4 different tasks. The grey trajectories are training demonstrations. The red trajectory represents the ground truth, while the blue, yellow, green, purple, and black trajectories represent the predictions by the TP-ProMP model, the ProMP model, the time-based TP-GMM model, the dynamic-based TP-GMM model, and the KMP model, respectively. The circles denote the start of the trajectories and the crosses denote the end of the trajectories. (a) Position predictions for the pick-and-place task. (b) Position predictions for the pouring water task. A square symbol is used to indicate approximately when the pouring action occurs. (c) Orientation predictions for the pouring water task. (d) Position predictions for the shooting puck task. (e) Position predictions for the sweeping task.

on unknown initial positions for grasping the pitcher at time $t = 0$ and an unknown via-point for pouring into the cup at an unknown time t. Conditioning the model on approximated position and time values is not only complex but also leads to poor performance. The TP-ProMP model was trained following the procedure outlined in Section II-E1 with $C = 2$. For more generality, the Bayesian information criterion can be used for model selection. Both TP-ProMP and dynamic-based TP-GMM compared predictions from different modes to the ground truth, selecting the prediction closest to the ground truth for evaluation. Figure 3b illustrates the performance comparison of the 5 models in position prediction for the water-pouring task.

For orientation prediction, we conducted a comparison between TP-ProMP, ProMP, time-based TP-GMM, and KMP models. In [28], the quaternions predicted by a time-based TP-GMM model were normalized to ensure they have unit norm. We followed the same procedure for all models with the exception of KMP. This step was unnecessary for KMP since it inherently respects the geometry of quaternions. Dynamic-

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE II
MEAN +/- STANDARD DEVIATION (SD) DISTANCE FROM GROUND TRUTH FOR PROMP, TP-GMM, KMP, AND TP-PROMP (GMM) FOR POURING WATER TASK.

	Position				Orientation
	Start (mm)	Mid (mm)	End (mm)	Average (mm)	Average
ProMP	249.6 ± 129.3	190.2 ± 88.96	140.6 ± 86.86	177.3 ± 74.83	1.028 ± 0.01667
TP-GMM(time-based)	113.2 ± 53.41	132.8 ± 94.44	176.3 ± 129.8	142.4 ± 95.78	0.6518 ± 0.3171
TP-GMM(dynamic)	78.78 ± 74.44	90.16 ± 68.36	121.9 ± 90.20	101.8 ± 69.68	N/A
KMP	92.78 ± 31.78	105.9 ± 79.48	184.5 ± 132.9	139.3 ± 92.54	1.025 ± 0.01694
TP-ProMP (GMM)	$\mathbf{18.83 \pm 8.181}$	$\mathbf{51.00 \pm 34.59}$	$\mathbf{83.97 \pm 52.92}$	$\mathbf{69.71 \pm 40.13}$	$\mathbf{0.1129 \pm 0.1112}$

based TP-GMM was not included in the comparison because the literature does not describe its application to orientation, and the creators' public code does not support orientation. Figure 3c provides a comparison of the performance of the 4 models in orientation prediction for the water-pouring task.

The average performance over 100 tests is summarized in Table II. TP-ProMP outperformed the other models in both position and orientation modeling, highlighting its strength in handling trajectories with multiple modes. In contrast to [29], which models position and orientation separately, our approach simultaneously captures both, eliminating the need to match separate position and orientation models. The robustness of the clustering used in TP-ProMP is a factor in its superior performance. As discussed in II-E1, using MAP estimate and estimating the uncertainty of $\boldsymbol{\omega}$ enables the proposed method to achieve a better estimate of $\boldsymbol{\omega}$. In this study, the proposed clustering algorithm achieved an accuracy of 93.5%, outperforming the algorithm presented [21], which achieved an accuracy of only 63.3%.

C. Shooting puck

Aiming is a more challenging task that has a low tolerance for variance. In order to enable our model to generalize to such tasks, we extended our method by assigning paired-object reference frames, as described in Section II-E2. The extended method was tested in a shooting puck task, where the demonstrator shoots a puck into a net on the table. In each demonstration, the position of the net remained fixed, while the position of the puck was randomly selected. The TP-ProMP was trained using trajectories in selected reference frames as discussed in Section II-E2, while TP-GMM and KMP models were trained on trajectories in object reference frames. The ProMP model, on the other hand, was trained using trajectories in the global reference frame.

Figure 3d presents a visual comparison between the 5 models in the shooting puck task. The models were trained on 6 demonstrations. As shown in the figure, the ProMP model performed poorly due to a lack of task parametrization. TP-GMM and KMP models, which incorporate object parameterization, demonstrate improved performance at the start and the end of the trajectory. However, the shooting action requires the gripper path to be straight in the puck-to-net direction for a successful outcome. The predictions generated by KMP and TP-GMMs exhibit significant curvature and deviate considerably from the ground truth. In contrast, TP-ProMP's predictions demonstrate great adaptation and exhibit minimal deviation from the ground truth, showing its superior performance in modeling the shooting action. Table III demonstrates the superior performance of TP-ProMP in general, except at the end of the trajectory where TP-ProMP shows slightly worse results compared to the KMP.

D. Sweeping

To further evaluate the effectiveness of the TP-ProMP method with paired-object reference frames, we conducted additional tests on another aiming task, specifically sweeping. In this task, a demonstrator cleaned rubbish into a dustpan. Due to physical constraints of the experimental setup, all demonstrations involved sweeping from the right to the left. To better mimic natural sweeping actions that can be executed in any direction, the demonstrations were augmented by rotating the trajectory and teabags' positions about the dustpan's vertical axis (z-axis) by a random angle. This augmentation produced a more diverse set of demonstrations.

The test results for the sweeping task are summarized in Table IV. Notably, TP-ProMP with paired-object reference frames exhibited significantly improved performance over the trajectory and at the start of the trajectory. Poorer performance at the trajectory's end (compared with the start) is attributed to the size difference between the dustpan and teabag. The teabags, being smaller, may land in various parts of the dustpan. While this variability causes greater distances from the ground truth at the trajectory's end, these distances are relatively small compared to the dustpan's width. Figure 3e shows a visual comparison between the 5 models. TP-ProMP with paired-object reference frames provides a prediction that deviates the least from the ground truth.

IV. CONCLUSION

Task-parameterized probabilistic movement primitives (TP-ProMP) offer a promising approach that combines the advantages of both ProMP and TP-GMM, making it well-suited for generalizing to different tasks. Its strengths include excellent extrapolation performance, simultaneous modeling of position and orientation, support for trajectories with multiple modes, and improved performance in aiming tasks. In future work we plan to study the method's efficiency (although it ran in $7-30$ms on a desktop computer), to compare this method to neural network (NN)-based methods such as those proposed in [15]–[19], and to integrate TP-ProMP into multi-step tasks.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE III
MEAN +/- SD DISTANCE FROM GROUND TRUTH FOR PROMP, TP-GMM, KMP, AND TP-PROMP FOR THE SHOOTING PUCK TASK.

	Start (mm)	End (mm)	Average (mm)
ProMP	269.8 ± 146.7	175.3 ± 92.73	237.0 ± 123.9
TP-GMM (time-based)	76.39 ± 46.93	55.24 ± 29.56	67.96 ± 24.32
TP-GMM (dynamic-based)	91.40 ± 79.73	79.72 ± 82.92	97.00 ± 86.21
KMP	71.99 ± 42.16	$\mathbf{51.76 \pm 29.42}$	65.87 ± 24.80
TP-ProMP	$\mathbf{45.28 \pm 25.08}$	52.92 ± 27.65	$\mathbf{57.57 \pm 24.38}$

TABLE IV
MEAN +/- SD DISTANCE FROM GROUND TRUTH FOR PROMP, TP-GMM, KMP, AND TP-PROMP FOR THE SWEEPING TASK.

	Start (mm)	End (mm)	Average (mm)
ProMP	638.1 ± 324.2	240.4 ± 111.9	474.9 ± 216.3
TP-GMM(time-based)	107.0 ± 58.87	189.2 ± 107.3	152.6 ± 64.19
TP-GMM(dynamic-based)	131.3 ± 122.8	289.3 ± 235.3	211.5 ± 125.5
KMP	97.62 ± 45.51	180.4 ± 95.67	139.3 ± 59.69
TP-ProMP	$\mathbf{68.98 \pm 61.91}$	$\mathbf{110.51 \pm 85.00}$	$\mathbf{106.45 \pm 49.03}$

ACKNOWLEDGMENTS

Supported by the Natural Sciences and Engineering Research Council of Canada and Applied Brain Research.

REFERENCES

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.

[2] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[3] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.

[4] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Few-shot learning: A survey," 2019.

[5] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive motion of animals and machines*. Springer, 2006, pp. 261–280.

[6] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[7] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," *Advances in neural information processing systems*, vol. 26, 2013.

[8] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, pp. 529–551, 2018.

[9] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2325–2332, 2023.

[10] N. Jaquier, D. Ginsbourger, and S. Calinon, "Learning from demonstration with model-based gaussian process," in *Conference on Robot Learning*. PMLR, 2020, pp. 247–257.

[11] M. Arduengo, A. Colomé, J. Lobo-Prat, L. Sentis, and C. Torras, "Gaussian-process-based robot learning from demonstration," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2023.

[12] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," 2015.

[13] M. J. Zeestraten, I. Havoutis, J. Silvério, S. Calinon, and D. G. Caldwell, "An approach for imitation learning on riemannian manifolds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1240–1247, 2017.

[14] Y. Huang, F. J. Abu-Dakka, J. Silvério, and D. G. Caldwell, "Generalized orientation learning in robot task space," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2531–2537.

[15] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, "Conditional neural movement primitives." in *Robotics: Science and Systems*, vol. 10, 2019.

[16] F. Xie, A. Chowdhury, M. De Paolis Kaluza, L. Zhao, L. Wong, and R. Yu, "Deep imitation learning for bimanual robotic manipulation," *Advances in neural information processing systems*, vol. 33, pp. 2327–2337, 2020.

[17] M. Przystupa, F. Haghverd, M. Jagersand, and S. Tosatto, "Deep probabilistic movement primitives with a bayesian aggregator," *arXiv preprint arXiv:2307.05141*, 2023.

[18] A. Tekden, M. P. Deisenroth, and Y. Bekiroglu, "Neural field movement primitives for joint modelling of scenes and motions," *arXiv preprint arXiv:2308.05040*, 2023.

[19] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," *arXiv preprint arXiv:2112.01511*, 2021.

[20] S. Haldar, J. Pari, A. Rai, and L. Pinto, "Teach a robot to fish: Versatile imitation from one minute of demonstrations," *arXiv preprint arXiv:2303.01497*, 2023.

[21] 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 International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1535–1542.

[22] V. R. Osorio, R. Iyengar, X. Yao, P. Bhattachan, A. Ragobar, N. Dey, and B. Tripp, "37,000 human-planned robotic grasps with six degrees of freedom," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3346–3351, 2020.

[23] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "Deeplabcut: markerless pose estimation of user-defined body parts with deep learning," *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.

[24] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[25] C. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING*, vol. 28, no. 2, pp. 623–635, 1980.

[26] S. Gomez-Gonzalez, G. Neumann, B. Scholkopf, and J. Peters, "Adaptation and robust learning of probabilistic movement primitives," *IEEE Transaction on Robotics*, 2017.

[27] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, pp. 155–164, 2009.

[28] J. Silvério, L. Rozo, S. Calinon, and D. G. Caldwell, "Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 464–470.

[29] L. Rozo and V. Dave, "Orientation probabilistic movement primitives on riemannian manifolds," in *Conference on Robot Learning*. PMLR, 2022, pp. 373–383.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.