2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Task Generalization with Pathwise Conditioning of Gaussian Process for Learning from Demonstration

Adrian Prados, Gonzalo Espinoza, Alberto Mendez and Ramon Barber.

***Abstract*— To effectively operate in human-centered environments, robots must possess the capability to rapidly adapt to novel and changing situations. Techniques such as Learning from Demonstration enable fast learning without the need for explicit coding. However, in certain cases they exhibit limitations in generalizing beyond the set of demonstrations, which constrains their ability to rapidly adapt to unforeseen scenarios. In this work, we present a movement primitive learning algorithm based on Gaussian Processes, combined with a zero-shot adaptation to new via-points without requiring retraining, through Pathwise Conditioning. The algorithm not only learns the movement policy but is also capable of adapting it rapidly while preserving prior knowledge. The method has been evaluated through comparisons against other state-of-the-art approaches, experiments in simulated environments, as well as on a real robotic platform, generating new solutions for learned tasks by modifying via-points in both position and orientation. Website project: `https://adrianprados.github.io/GaussianPathwiseLfD/`.**

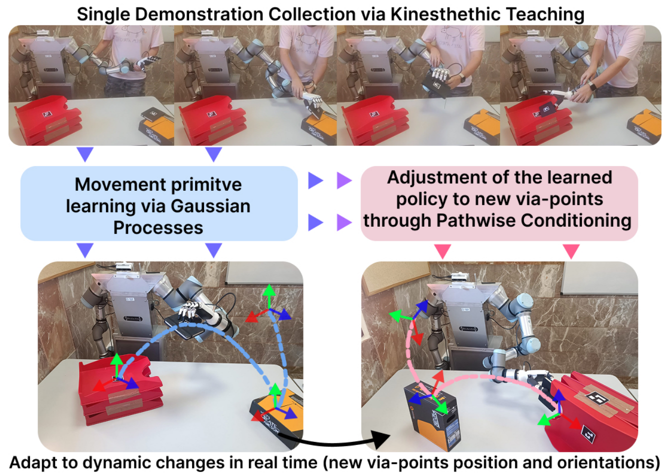

Fig. 1. Using a single demonstration, movement primitives are learned via GPs and adapted in real time to new via-points using Pathwise Conditioning.

I. INTRODUCTION

In recent years, robots have been increasingly introduced into everyday environments, which are often unstructured. For robots to operate correctly, it is essential that they are capable of adapting to these changing environments, working in a reactive and safe manner, and being easy to program [1], [2]. Robots need to constantly acquire new skills in order to meet the evolving and dynamic tasks of their surroundings. Such adaptation can be complex to program, not only for non-experts but also for experts alike. One way to endow robots with the ability to learn new skills is through the use of Learning from Demonstration (LfD) algorithms [3], [4]. This technique focuses on providing robots with the capability to acquire new skills by encoding them into a set of policies. These policies enable robots to learn actions or movements demonstrated by a teacher, often a human, thereby equipping them with abilities similar to ours. This technique has been gaining increasing relevance in recent years as a means to quickly acquire complex skills. However, many of the approaches developed focus on learning and executing tasks in environments and contexts without variations. This can lead to failures when working in new scenarios or situations that lie outside the learned distributions, the out-of-distribution (OOD) problem. Within OOD, *covariant shift* [5], [6] is one of the classical limitations of LfD. By relying on a fixed set of training data, LfD algorithms learn a policy that performs optimally within those data and environments, but may fail when attempting to generalize to unseen inputs during the training process. Consequently, this distribution shift renders the learned policy invalid. To address this limitation, the robot should not merely memorize the demonstrations for a specific scenario, but instead be able to generalize and adapt its knowledge to new situations (even in real-time) while pursuing similar task objectives. Within LfD, a large number of algorithms have been developed based on probabilistic techniques, such as Gaussian Mixture Models (GMM) [7], [8], Gaussian Mixture Regression (GMR [9], [10], Dynamical Movement Primitives (DMP) [11], [12], Probabilistic Movement Primitives (ProMP) [13], [14], Kernelized Movement Primitives (KMP) [15], [16], or Gaussian Process Regression (GPR) [17], [18], which aim not only to learn the task but also to adapt such learning to *covariant shift* situations. Methods based on GMM/GMR, KMPs, and ProMPs focus on using the provided demonstrations to encode variability as a function of the dispersion of the data. In contrast, methods based on GPRs focus on using covariance matrices based on prediction uncertainty, reflecting the presence or absence of data in the task. Both techniques allow estimating new solutions beyond the initial demonstrations, but they do not take each other into account. That is, methods based on variability do not consider the availability of data, which necessitates data augmentation or additional techniques to handle requirements not captured in the initial demonstrations, while prediction-based methods are not capable of learning the variability of the data.

In this paper, and considering both, prediction uncertainty and variability, we have developed a new technique (Fig-

All authors of the paper are with the same institution RoboticsLab, University Carlos III, 28911 Leganés, Spain. {aprados,gespinoz,albmende,rbarber}@ing.uc3m.es

ure 1) based on the use of Gaussian Processes (GPs) [19], [20] for learning from the initial demonstration data, combined with a subsequent correction based on Pathwise Conditioning [21], [22]. By leveraging prior knowledge from the demonstrations, this approach is capable of correcting the trajectory in real time—capturing both uncertainty and variability—so that the task can be successfully executed in new situations, both in orientation and position. The proposed method is capable of adapting its learned motion policy from one or multiple demonstrations to previously unknown situations without the need to generate new data, at 30Hz, allowing online adaptation even when the robot or the relevant parameters of the task are moving.

II. RELATED WORK

Within LfD, the number of methods developed to solve complex tasks and adapt them to new situations, aiming to address the limitations of out-of-distribution tasks, is vast [23]–[25]. Specifically, within the probabilistic algorithms used for LfD, different methods and strategies have been developed. Previous work in this field has focused on using multitask learning techniques [26], [27], training a single model that learns several related tasks jointly, leveraging shared information among them to improve individual generalization. Other techniques are based on applying meta-learning [28], [29], which focuses on learning the task and, from just a few demonstrations, generalizing the knowledge already acquired to the new task to be performed. Although these methods are capable of generalizing learned tasks to new situations or environments, they require a large amount of data and online or offline training and re-training. This large data requirement prevents them from being used reactively in task execution, making them difficult to apply for real-time corrections. One of the techniques that has been most widely used in recent years is task generalization through the use of *task parametrization* (TP) [30]. This idea seeks to represent tasks by means of a set of task frames, which are usually contextual elements (objects or key points of the task). Instead of encoding trajectories exclusively in global coordinates, each task is represented from these task frames. Then, when executing the task in new configurations, the robot fuses these representations conditioned on the new state of the environment. Within this technique, the use of algorithms based on Task-Parameterized Gaussian Mixture Models (TP-GMM) [31]–[34], Task-Parameterized Dynamic Movement Primitives (TP-DMP) [35], [36], or approaches based on Gaussian Processes (TP-GP) [37], [38], has stood out. The efficiency of these methods has been clearly demonstrated in different areas and robotic tasks; however, task parameterized methods suffer from some limitations when working with few demonstrations, since they require several parameters that has to be well distributed. These methods are also sensitive to the choice of irrelevant or even redundant frames and have certain limitations when dealing with real-time problems or tasks with sharp directional changes. Other works focus on directly learning motion policies. In these methods, the aim is to explicitly adapt the parameters of such policies. Among these techniques, methods based on Probabilistic Movement Primitives (ProMPs) [39], [40] stand out, through adaptations of relevant task points or descriptors that generate geometric solutions. Other approaches combine motion policies learned through probabilistic techniques such as GMM with learning of Dynamical Systems (DS) [41], [42], which allows tasks to be adapted quickly, with one-shot learning, while ensuring stability.

In this work, we propose a new LfD method based on a probabilistic technique, specifically Gaussian Processes, for learning tasks from one or multiple demonstrations. Our algorithm not only aims to learn the structure of the tasks but also seeks to address the problems of out-of-distribution data and adaptation to new tasks. To this end, instead of performing classical regression with methods, we employ the technique of Pathwise Conditioning, which, by avoiding computations with covariance matrices, allows for fast adaptation to a large number of new task parameters. This generates a solution that leverages the initial knowledge but adapts it using a zero-shot approximation, without the need to add or synthetically generate new data.

III. PROBLEM STATEMENT

A. Gaussian Process

Gaussian Processes are a collection of random variables, where any finite subset of variables is jointly distributed according to a Gaussian distribution. Intuitively, they can be seen as a distribution over a function $f(x)$ such that $f : \mathbb{R}^d \rightarrow \mathbb{R}$. For any finite dataset $x_1, x_2, \ldots, x_n \subset \mathbb{R}^d$, one can construct a list of function evaluations $[f(x_1), f(x_2), \ldots, f(x_n)]$, which follows a multivariate normal distribution. A GP is defined by the mean of the data $m(t) = E[f(t)]$ and its covariance function $k(t, t') = E[(f(t) - m(t))(f(t') - m(t'))]$, which encodes the uncertainty, where $f(t)$ denotes the stochastic functions. GPs incorporate prior knowledge $D = (t_k, q_{k,i})k, i = 1^N$ on $f(t)$, where t_k denotes the time of each element and qk, i the i-th dimension for a given t_k. Each $q_{k,i}$ is generated as $q_{k,i} = h_i(t_k) + \epsilon_{k,i}$, with only noisy observations being available. $\epsilon_{k,i}$ represents observation noise, which is assumed to follow a Gaussian distribution $\mathcal{N}(m, \sigma^2_{no,i})$. The latent function is also modeled as a GP, $h_i(t^) \sim \mathcal{GP}(m_i(t^*), k_i(t, t'))$. Hence, the output vector $q_i = [q_{1,i}, \ldots, q_{N,i}]^\top \in \mathbb{R}^N$ follows a multivariate Gaussian distribution. Given a GP prior and (noisy) observations, the joint law of observed targets q and latent values f^* at test inputs is Gaussian. Evaluating the prior mean and covariance on the training inputs t yields $\mathbf{m}(t)$ and $K(t, t^*)$. Accounting for observation noise modelled as zero-mean, uncorrelated Gaussian errors with covariance $\sigma_n^2 I$, the joint distribution becomes [37]

$$\begin{bmatrix} q \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}(t) \\ \mathbf{m}(t^*) \end{bmatrix}, \begin{bmatrix} K(t,t) + \sigma_n^2 I & K(t,t^*) \\ K(t^*,t) & K(t^*,t^*) \end{bmatrix} \right) \quad (1)$$

Consequently, each observation vector q_i is distributed as $p(q_i \mid t) = \mathcal{N}\big(\mathbf{m}_i(t),\ K_i(t,t) + \sigma^2_{n,i} I\big)$. A very important elemental in the GPs is the kernel $k_i(\cdot, \cdot)$, that embeds the GP hyperparameters θ_i (e.g. $\theta_i = \{\sigma_{ker,i}, l_i\}$). There is a wide variety of kernels that can be used with GPs, but one of the most commonly used, due to its simplicity and versatility in use cases, is the squared–exponential kernel $k_i(t,t') = \sigma^2_{ker,i}\exp\Big(-\frac{(t-t')^2}{l_i^2}\Big)$, whose hyperparameters θ_i are obtained by maximizing the log marginal likelihood $\log p(q_i \mid t) = -\frac{1}{2}q_i^\top\big(K(t,t)+\sigma_n^2 I\big)^{-1}q_i - \frac{1}{2}\log\det\big(K(t,t)+\sigma_n^2 I\big) - \frac{n}{2}\log 2\pi$. In this work, kernel selection (and hyperparameter tuning) is automated via Approximate Bayesian Computation (ABC) [43] in order to avoid manual specification. Given optimized hyperparameters, the predictive distribution for $h_i(t^*) \in \mathbb{R}^M$ at test inputs $t^* \in \mathbb{R}^M$ is Gaussian:

$$p(h_i(t^*) \mid t^*, D_i) = \mathcal{N}\big(\mu_i(t^*, D_i), \Sigma_i(t^*, D_i)\big), \quad (2)$$

with posterior mean and covariance given by [37]:

$$\begin{cases}\mu_i(t^*, D_i) = \mathbf{m}(t^*) + K_i(t^*,t)(K_i(t,t) + \sigma^2_{no,i}I)^{-1}(q_i - \mathbf{m}(t)),\\ \Sigma_i(t^*, D_i) = K_i(t^*,t^*) - K_i(t^*,t)(K_i(t,t) + \sigma^2_{no,i}I)^{-1}K_i(t,t^*).\end{cases} \quad (3)$$

B. Pathwise Conditioning

Conditioning stochastic processes is a fundamental task in machine learning and its application to robotics. Traditionally, this conditioning is approached from a distributional perspective, where the goal is to characterize the distribution of a random variable $\mathbf{a}$ given that another variable $\mathbf{b}$ takes a specific value $\boldsymbol{\beta}$, i.e., $p(\mathbf{a} \mid \mathbf{b} = \boldsymbol{\beta})$. This approach may present limitations when it is necessary to sample from conditional distributions in order to obtain actionable estimates, particularly in complex scenarios. Pathwise conditioning emerges as an alternative, offering a perspective complementary to the distributional approach. This method focuses on the behavior of a GPs at a finite number of input locations, allowing GPs posteriors to be decomposed into global and local components. The idea is to generate efficient function samples from GPs posteriors, simplifying interactions with complex mathematical expressions and providing deeper insights into the contributions of the prior and the data. Instead of sampling after conditioning, the pathwise approach proposes sampling before conditioning [22]. The key mathematical foundation of pathwise conditioning for jointly Gaussian variables is Matheron's Update Rule [44]. This theorem states that if two random vectors $\mathbf{a}$ and $\mathbf{b}$ are jointly Gaussian, then the conditional random variable $\mathbf{a}$ given $\mathbf{b} = \boldsymbol{\beta}$ can be expressed as the sum of two independent terms: the conditional expectation $E[\mathbf{a} \mid \mathbf{b} = \boldsymbol{\beta}]$ (evaluated at $\boldsymbol{\beta}$), and a residual $\mathbf{c} = \mathbf{a} - E[\mathbf{a} \mid \mathbf{b}]$ which is independent of $\mathbf{b}$. Formally, $(\mathbf{a} \mid \mathbf{b} = \boldsymbol{\beta}) \overset{d}{=} \mathbf{a} + \Sigma_{\mathbf{a},\mathbf{b}}\Sigma^{-1}_{\mathbf{b},\mathbf{b}}(\boldsymbol{\beta} - \mathbf{b})$, where $\mathbf{a}$ on the right-hand side represents a random draw from the prior, and $\Sigma_{\mathbf{a},\mathbf{b}}\Sigma^{-1}_{\mathbf{b},\mathbf{b}}(\boldsymbol{\beta} - \mathbf{b})$ is a deterministic update based on observations. Applied to Gaussian processes—random functions $f : \mathbf{X}_n \to \mathbb{R}$ where any finite set of points follows a joint Gaussian distribution—pathwise conditioning provides an efficient way to update sample paths. For $f \sim \mathcal{GP}(0,k)$ with marginal $f_n = f(\mathbf{X}_n)$ conditioned on $f_n = \mathbf{y}$:

$$f(\cdot) \mid f_n = \mathbf{y} \overset{d}{=} f(\cdot) + k(\cdot, \mathbf{X}_n)K_n^{-1}(\mathbf{y} - f_n), \quad (4)$$

where $f(\cdot)$ is a prior GP, and $k(\cdot, \mathbf{X}_n)K_n^{-1}(\mathbf{y} - f_n)$ is the deterministic pathwise update. This separates the contributions of the prior and the data, generating efficient and accurate posterior samples—essential for applications requiring fast, reliable estimates.

IV. PROPOSED METHOD

We present our algorithm for task learning using GPs and generalization through a zero-shot method based on Pathwise Conditioning. First, the LfD process is carried out using GPs to learn movement primitives, which may include initial via-points to adapt to. After this, an adaptation to new positions and orientations of either known or new via-points is performed through Pathwise Conditioning. The method produces fast solutions that exploit the initial information by combining the learned policy with a new local policy for each situation, thereby adjusting the overall task where the re-learning process is avoided.

A. LfD using Gaussian Process

For this work, we have developed an LfD method based on the encapsulation of movement primitives through GPs. To achieve this, we have introduced a modification of the extended use of GPs in order to encapsulate movement primitives [45] in a non-parametric way, adding the capability of automatic kernel adaptation to different movement primitives. The proposed method is fully data-driven and explicitly encodes predictive uncertainty at each time instant, which enables confidence-dependent control decisions. Moreover, the GP structure facilitates the incorporation of via-point constraints in the initial learning process. This allow an initial approximation that directly accounts for the relevant task points before being modified or adapted to new tasks.

The algorithm begins with data acquisition from the user. In this work, kinesthetic data is used, and both a set of demonstrations or a single demonstration can be employed. The available demonstrations are used to create an initial heteroscedastic approximation. This allows initialization of the mean and an initial training kernel before training the GP. Instead of assuming homogeneous noise, we propose a heteroscedastic initialization that adapts the noise structure to the variability observed in the demonstrations. Figure 2 presents an example in 3D when applying our LfD method. The procedure follows these steps:

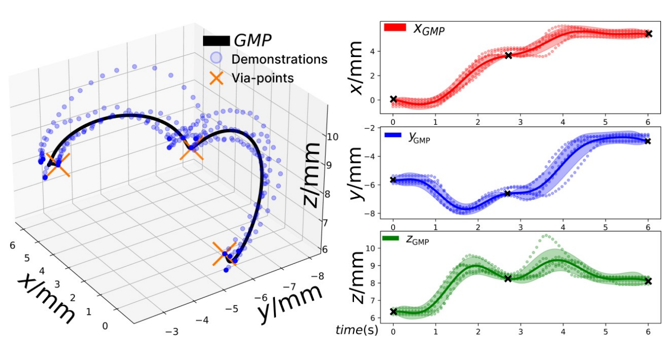

Fig. 2. Solution of movement primitives learning via GPs in 3D with 3 via-points. The GP decomposition for each dimension is shown, where the shaded area represents the covariance; at the via-points the passage is 100% certain (no shaded area is present).

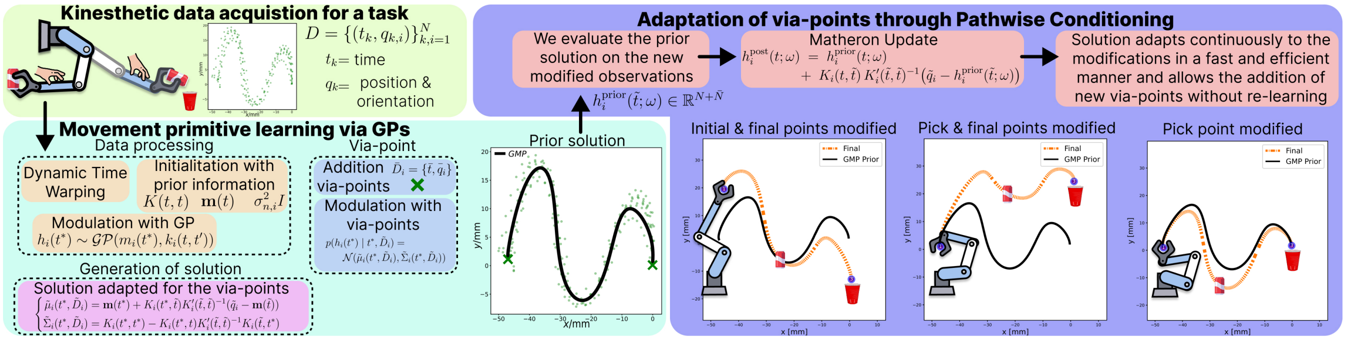

Fig. 3. General scheme. First, kinesthetic data are acquired (time, position/orientation) and preprocessed to temporally align and encapsulate the demonstrations. Next, movement primitives are learned via Gaussian Processes: a heteroscedastic initialization of the mean and noise is computed from the demonstrations' envelope, an initial kernel is constructed, and hyperparameters are optimized automatically, yielding a prior over trajectories. Using this prior knowledge, a linear correction is applied via Pathwise Conditioning to the prior trajectory to enforce new via-points without retraining, locally preserving the learned primitive and its predictive uncertainty.

I) Filtering and Envelope. From the user data y, a dynamic envelope (bounds lb, up) is estimated to exclude outliers and select points within the region of interest $y_{\text{in}} = \{y_i \mid lb_i \leq y_i \leq up_i, \forall i\}$. When more than one demonstration is used, a Dynamic Time Warping (DTW) [46] process is first applied to temporally align all demonstrations, as otherwise the encapsulation may fail.

II) Local Statistics. The mean and standard deviation over y_{in} are computed by $\mu_{\text{in}} = \frac{1}{N}\sum_{i=1}^N y_{\text{in},i}$ and $\sigma_{\text{in}} = \sqrt{\frac{1}{N}\sum_{i=1}^N (y_{\text{in},i} - \mu_{\text{in}})^2}$ respectively. With this, a diagonal noise matrix is defined $R(t) = \sigma^2_{\text{in}} I_{y_{\text{in}}}$.

III) Heteroscedastic Initial Kernel. A base initial kernel (e.g., squared-exponential) is defined between $m(t) = \mu_{\text{in}}$:

$$K_{\text{base}}(i,j) = \exp\Big(-\frac{(m_i(t) - m_j(t))^2}{\ell_i^2}\Big) \tag{5}$$

The initial kernel is constructed from the mean $m(t)$ and the locally estimated variance from the filtered data $K_{\text{init}} = R(t)^2 \odot K_{\text{base}}$, where $\odot$ represents the Hadamard product. With this definition, the observed variability at each point locally scales the covariance: entries corresponding to indices i with larger dispersion in the demonstrations acquire greater magnitude in the initial covariance matrix, thus reflecting the heteroscedasticity of the data. To preserve symmetry and numerical stability, one can alternatively consider $K_{\text{init}} = DK_{\text{base}}D$, with $D = \sigma_{\text{in}} I_{y_{\text{in}}}$. It is advisable to add a jitter term ϵI if necessary, to ensure that the matrix is positive definite and Cholesky-factorizable, thus avoiding problems in matrix inversion.

IV) Automatic Kernel Selection. Once we have K_{init}, we aim to obtain $K_i(t,t)$. For this purpose, we make use of a method based on Approximate Bayesian Computation (ABC) [43]. The objective of this method is to obtain the most computationally suitable solution in terms of efficiency. To this end, the optimization problem is defined as:

$$\pi(K_{\text{init}} \mid t^*, G) \propto f(t^* \mid K_{\text{init}}, G)\, \pi(K_{\text{init}} \mid G) \tag{6}$$

where G represents the model based on the kernel functions K_{init}, $\pi(K_{\text{init}} \mid G)$ corresponds to the initial distribution obtained through a heteroscedastic initialization process grounded on the structure of the demonstration data, and $f(t^* \mid K_{\text{init}}, G)$ represents the reliability of the observations, denoted as t^*, for the initial kernel function K_{init}. The ABC algorithm compares models and discards those with a lower probability of converging to an optimal solution. In this way, the algorithm not only achieves a kernel that accurately fits the data, but also automatically optimizes the kernel hyperparameters, thus removing the need for an explicit likelihood optimization process. Once $K_i(t,t)$ is obtained, each value is multiplied by a weight difference obtained as the product of the standard deviation considering the difference between the limits. This allows for significant differences between limit values to be reflected in this optimal kernel.

After obtaining the optimized kernel for the human demonstrations, the algorithm needs to generate solutions that pass through a series of initial via-points. An important advantage of movement primitives over other GP-based methods is the possibility of enforcing that the trajectory goes through via-points. To ensure that our GP-based method strictly meets these requirements, we incorporate the via-points into the training set instead of adding them only as a post-condition. Let $D_i = \{t, q_i\} \in \mathbb{R}^N$ be the set of demonstrations (time and points), and $\bar{D}_i = \{\bar{t}, \bar{q}_i\} \in \mathbb{R}^{\bar{N}}$ be the set of $\bar{N}$ via-points (assumed to be exact, noiseless). We define the union $\tilde{D}_i = \{D_i, \bar{D}_i\}$, $\tilde{t} = \begin{bmatrix} t^T & \bar{t}^T \end{bmatrix}^T \in \mathbb{R}^{N+\bar{N}}$, $\tilde{q}_i = \begin{bmatrix} q_i^T & \bar{q}_i^T \end{bmatrix}^T \in \mathbb{R}^{N+\bar{N}}$. The joint distribution over observations and via-points is:

$$p\left(\begin{bmatrix} q_i \\ \bar{q}_i \end{bmatrix} \Big| \begin{bmatrix} t \\ \bar{t} \end{bmatrix}\right) \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}(t) \\ \mathbf{m}(\bar{t}) \end{bmatrix}, \begin{bmatrix} K(t,t) + R(t) & K(t,\bar{t}) \\ K(\bar{t},t) & K(\bar{t},\bar{t}) \end{bmatrix}\right) \tag{7}$$

Using the optimized hyperparameters, the prediction for a test point t^* with the GP incorporating via-points is

$$p(h_i(t^*) \mid t^*, \tilde{D}_i) = \mathcal{N}\big(\tilde{\mu}_i(t^*, \tilde{D}_i),\, \tilde{\Sigma}_i(t^*, \tilde{D}_i)\big), \text{with} \tag{8}$$

$$\begin{cases} \tilde{\mu}_i(t^*, \tilde{D}_i) = \mathbf{m}(t^*) + K_i(t^*, \tilde{t}) K_i'(\tilde{t}, \tilde{t})^{-1} (\tilde{q}_i - \mathbf{m}(\tilde{t})) \\ \tilde{\Sigma}_i(t^*, \tilde{D}_i) = K_i(t^*, t^*) - K_i(t^*, t) K_i'(\tilde{t}, \tilde{t})^{-1} K_i(\tilde{t}, t^*) \end{cases} \tag{9}$$

where the kernel value with the via-points $K_i'(\tilde{t}, \tilde{t})$ is:

$$K_i'(\tilde{t}, \tilde{t}) = \begin{bmatrix} K_i(t,t) + R(t) & K_i(t,\bar{t}) \\ K_i(\bar{t},t) & K_i(\bar{t},\bar{t}) \end{bmatrix} \tag{10}$$

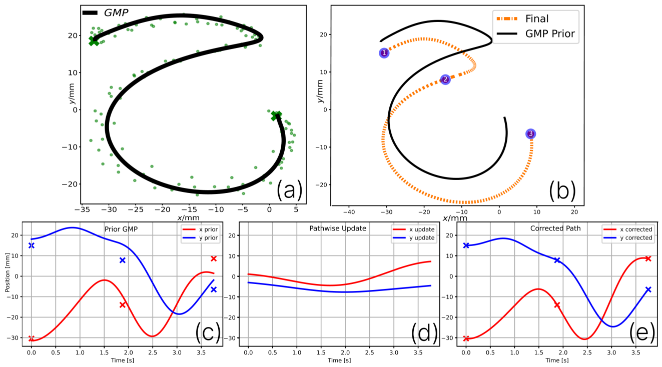

Fig. 4. GPs movement primitive learning with pathwise update in 2D. (a) Prior solution with 2 via-points, (b) conditioned solution with 3 via-points in new positions, (c) prior GP for each dimension, (d) correction performed through Pathwise Conditioning to enforce passage through the new via-points, (e) final solution combining prior knowledge with the adapted component.

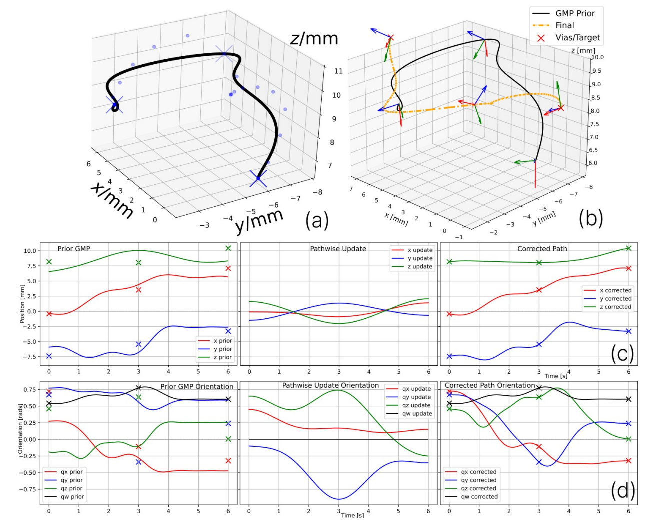

Fig. 5. GPs movement primitive learning with pathwise update in 3D (position + quaternions). (a) Prior solution with 3 via-points and 1 demonstration, (b) conditioned solution with 3 via-points in new positions and orientations (axes with arrows), (c) correction of prior knowledge through pathwise update in position, (d) correction of prior knowledge through pathwise update in orientation.

It is essential that $K'_i(\tilde{t},\tilde{t}) \in \mathbb{R}^{(N+\bar{N})\times(N+\bar{N})}$ is invertible in order to use (9). Applying known results on kernels [47], at least positive semidefiniteness is guaranteed. To ensure a numerically stable inverse, it is advisable to check the invertibility condition and, if necessary, add a stabilization term (jitter) ϵI. With these precautions, the GP incorporating $\tilde{D}_i$ guarantees that predictions pass exactly through the included via-points. For formal details on sufficient conditions ensuring exact passage through via-points, see [48].

B. Adaptation to new situations via Pathwise Conditioning

Once we have computed the GP, we obtain a policy that models the movement primitive generated from the demonstration data and that contains the initial via-points, for which the probability of passage is 100%, since they are included as training data. The next goal is to adapt in real time when the via-points change. For this process, GP-based methods that rely on GPR suffer, since they must recompute the entire regression due to the modified observations. This requires recalculating the matrix inverses, which is computationally expensive. Our previously described method instead defines a prior GP over the trajectories, so it is not necessary to retrain from scratch; it is only required to condition the GP on those specific points. This is a more local computation that avoids relearning the entire model. Although this approach is more efficient, it is still not fast enough to generate real-time solutions. This is why Pathwise Conditioning was adopted. The conditioning of the via-points is applied directly on the pathwise-sampled trajectory, not on the global model. This means that nothing from the original GP needs to be recomputed. The via-points are imposed directly on the learned trajectory, which is then combined with the prior knowledge in a zero-shot manner. An example of our method implementing this idea is presented in Figure 4. We adopt a pathwise conditioning strategy to adapt learned movement primitives via GPs to new or modified via-points that may lie outside the distribution of the original demonstrations. The method constructs corrected trajectories by applying a linear Matheron-style update to an existing prior trajectory (either the GP mean or a sampled GP path). This provides an efficient, constructive way to enforce via-point constraints while preserving the probabilistic semantics of the GP posterior. Starting from a prior trajectory $h_i^{\text{prior}}(\cdot;\omega)$, obtained through the deterministic posterior mean $\mathbf{m}(\cdot)$ after training the model with GPs or as a stochastic sample drawn from the prior GP via an appropriate weight-space or basis-function expansion, the prior trajectory is evaluated at the observation times $h_i^{\text{prior}}(\tilde{t};\omega) \in \mathbb{R}^{N+\bar{N}}$. The pathwise-conditioned is defined, for every realization ω:

$$h_i^{\text{post}}(t;\omega) \;=\; h_i^{\text{prior}}(t;\omega) \;+\; K_i(t,\tilde{t})\,K'_i(\tilde{t},\tilde{t})^{-1}\big(\tilde{q}_i - h_i^{\text{prior}}(\tilde{t};\omega)\big) \quad (11)$$

Equation (11) is the pathwise (Matheron) update: it modifies the prior trajectory by a data-dependent linear correction so that the updated trajectory passes through the augmented observations $\tilde{q}_i$. When the prior is a sample from the prior GP, the collection $\{h_i^{\text{post}}(\cdot;\omega)\}_\omega$ are distributed exactly according to the GP posterior conditioned on $\tilde{D}_i$. When the prior is the deterministic mean $\mathbf{m}(\cdot)$ the update recovers the posterior mean:

$$\tilde{\mu}_i(t) = \mathbf{m}(t) + K_i(t,\tilde{t})\,K'_i(\tilde{t},\tilde{t})^{-1}\big(\tilde{q}_i - \mathbf{m}(\tilde{t})\big) \quad (12)$$

For this optimization, we employ a separate independent kernel with one kernel per output coordinate, optimizing them separately and thus solving different one-dimensional problems. Under this choice, the update (11) decouples by output dimension d:

$$\Delta f_d(t) = K^{(d)}(t,\tilde{t})\big(K^{(d)}(\tilde{t},\tilde{t}) + \Sigma^{(d)}_{\text{obs}}\big)^{-1}\big(\tilde{q}_{i,d} - h_d^{\text{prior}}(\tilde{t})\big) \quad (13)$$

and the final corrected trajectory is $h_d^{\text{post}}(t) = h_d^{\text{prior}}(t) + \Delta f_d(t)$. If cases where output correlations are important, a multi-output kernel can be employed in the pathwise process and the update applied with the corresponding block covariance.

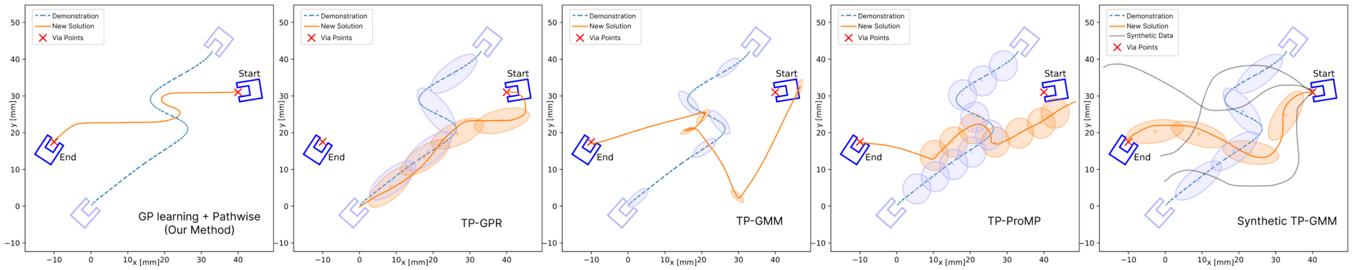

Fig. 6. Original demonstration shown as blue. New motion policy shown as orange. Ellipses represent probabilistic encoding. The compared methods do not satisfy the constraints when given a single demonstration or need synthetic data to achieve good solutions. Our method can generalize to the constraints imposed by the geometric descriptors at both endpoints in position and orientation. Other comparisons are presented in our WebPage.

With this method, in cases where the provided via-points $\bar{q}_i$ lie outside the distribution of the demonstration data, the pathwise update corrects the prior trajectory so that it passes through the new constraint, while minimally perturbing the trajectory in regions far from the observation times. The linear correction is weighted by the kernel cross-covariances $K_i(t, \tilde{t})$: for test times t remote from $\tilde{t}$, the correction tends to zero, preserving the learned primitive elsewhere, which maintains the structure of the previously learned movement whenever possible. This property, together with computational efficiency, makes pathwise conditioning suitable for the interactive adaptation of primitives under abrupt or out-of-distribution target changes.

One of the strengths of learning movement primitives through GPs is the ability to work in multiple dimensions. For this reason, the developed algorithm can efficiently handle orientation information during a task, allowing solutions to adapt not only to pass through specific via-points but also to do so with specific orientations. Figure 5 presents an example of this application: a 3D task containing both orientation information (axes with arrows) and position in the demonstration is learned through GP encapsulation to pass through initial points, and is subsequently corrected via pathwise update to modify those via-points, ensuring the trajectory passes through the new via-points with the corresponding position and orientation constraints. Therefore, the method enables rapid task correction using full spatial constraints in both position and orientation.

V. SIMULATION AND EXPERIMENTS

To evaluate the efficiency of our method, we conducted both quantitative and qualitative experiments. Tests were conducted against several state-of-the-art methods to evaluate its performance. In addition, we carried out different efficiency evaluations in real tasks using a IIWA arm and our robot ADAM [49], [50], both in simulation through PyBullet [51] and in a real-world environment.

A. Comparison against other methods in 2D

To evaluate the efficiency of our method, we conducted a comparison against four different task generalization algorithms for both position and orientation, based on task parameters: TP-GPR [30], TP-GMM [33], TP-ProMP [13], and Synthetic-TP-GMM [31]. The experiments rely on a single demonstration with constraints only at the endpoints. Both endpoints (blue U-Shape polygons) are modified in position and orientation to illustrate different configurations of the solutions produced by each method. Several experiments were performed using a single demonstration while altering the endpoint constraints. The tests are categorized as *final restriction near*, *final restriction far*, *both restrictions near* and *both restrictions far*, with orientation modified in all cases. Figure 6 presents the solutions of the comparison among all methods under *both restrictions far* case.

TABLE I
COMPARISON OF METRICS BETWEEN DIFFERENT METHODS

Method	Start Cos.	End Cos.	End. Dist.	Time(s)
Ours	**0.9997**	**0.9999**	**0.0001**	**0.0280**
TP-GPR	-0.9012	0.6987	0.4698	0.4781
TP-GMM	0.9125	0.513	0.7769	0.0509
TP-ProMP	0.2048	0.8799	0.4125	0.0476
Syn. TP-GMM	0.9224	0.9012	0.0008	2.0190

To perform the quantitative comparison, we defined several metrics [41] to evaluate the generalization capability (Equations (14)). *Start Cosine Similarity* ($\cos(\theta_s)$) allows us to describe the initial direction of the trajectory and its alignment with the U-Shape descriptor at the entry, through the relation between the starting trajectory vector v_s and the entry vector of the U-Shape v_o. The closer this value is to 1, the more optimal the alignment. *Goal Cosine Similarity* ($\cos(\theta_g)$) establishes the same criterion as the Start Cosine Similarity but for the trajectory's arrival. For this, we obtain the vector v_g and compare it with the exit vector of the U-Shape v_e. The closer this value is to 1, the more optimal the alignment. Finally, we use *Endpoint Distance*(D), which evaluates whether both the starting and ending points of the trajectory reach the required endpoints without position error. This metric relies on the positions of the initial and final U-Shapes (P_s, P_e) and the initial and final trajectory values (ξ_s, ξ_g). The smaller this value, the better. The quantitative results of Figure 6 are presented in Table I.

$$\cos(\theta_s) = \frac{v_s \cdot v_o}{\|v_s\| \, \|v_o\|}, \cos(\theta_g) = \frac{v_g \cdot v_e}{\|v_g\| \, \|v_e\|}, \quad D = d(\xi_s, P_s) + d(\xi_g, P_e) \tag{14}$$

As can be observed both qualitatively and quantitatively, the TP-GPR, TP-GMM, and TP-ProMP algorithms are unable to generalize from a single demonstration to new positions and orientations at both the beginning and the end of the trajectories. The TP-GPR algorithm suffers from generalization issues at the start, obtaining negative values in the Start Cos. metric (indicating that it begins with an orientation opposite to the required one) and producing a very large error in the end distance. The TP-GPR and TP-ProMP methods show serious generalization errors: the first is unable to generate a consistent and executable solution, while the second suffers from errors both in the trajectory and in the position of the starting point. Finally, the Synthetic TP-GMM algorithm is capable of generating correct solutions, approaching the efficiency of the algorithm presented in this work, since it synthetically generates additional demonstrations to expand the dataset it learns from. Despite this, the algorithm does not properly account for the initial learning structure, as its influence diminishes when the number of demonstrations increases, and it also faces execution time issues, requiring more than 2 seconds to generate a new solution, which makes it inefficient for tasks in real dynamic environments. The rest of the experiments and their quantitative results are detailed on the project's website[1].

B. Robot Experiments

To carry out the experiments with the robots, both simulation environments in PyBullet and real-world tests were used (Figure 7). In the simulation environments, a Kuka IIWA robotic arm and a mobile manipulator robot were employed (the latter also used in the real-world experiments). For the tasks, a single demonstration was used for learning. Relevant via-points were detected using visual tags captured with OpenCV. The experiments were conducted on an MSI Katana GF66 with an Intel Core i7 processor (4.7 GHz) and a Nvidia GeForce RTX 3070 GPU.

Data collection was carried out using the end-effector position p, orientation R, and task time t. The demonstrations are encapsulated in $q = [p, R]$ together with time, $D = \{t, q\} \in \mathbb{R}^N$. The visual tags represent via-points with new positions and orientations $\bar{D} = \{\bar{t}, \bar{q}\} \in \mathbb{R}^{\bar{N}}$. After this, the learning process of movement primitives is performed using GPs. This task incurs higher computational cost as the resolution of the demonstrations, the number of demonstrations, and the dimensionality increase. In these experiments, for 3D tasks with one demonstration of 500 points and dimension $N = 7$, the initial policy is obtained in an average of 0.89s. Afterwards, the algorithm retains the initial policy, which serves as the learning basis to subsequently apply the necessary corrections according to the new via-points' positions and orientations through optimization based on Pathwise Conditioning. For the real-world environment, different tasks simulating everyday activities were conducted, such as placing books on shelves, pick-and-place operations, liquid serving tasks, and testing the algorithm's ability to adapt the trajectory under direct interferences with the robotic arms (e.g., moving the arm to a new point during execution). In all cases, the algorithm was able to generate fast, efficient, and adaptive solutions to accomplish the initially learned task after modifications of the goals[1].

[1]https://adrianprados.github.io/GaussianPathwiseLfD/

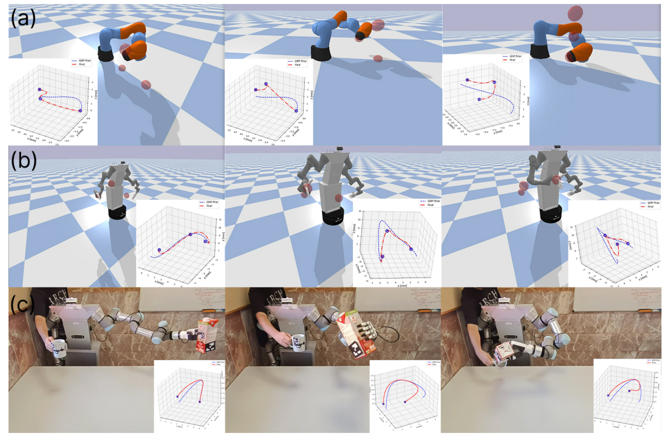

Fig. 7. Learning method of primitives using GPs + Pathwise Conditioning applied to a simulated KUKA arm (a), ADAMSim [51] (b), and in a real environment with ADAM robot (c). Red spheres represent the via-points modified in position and orientation, blue lines indicate the GP prior, and red lines show the real-time adapted solutions.

VI. CONCLUSIONS

In this paper, we introduce an algorithm for learning movement primitives using Gaussian Processes (GPs) from one or multiple demonstrations, combined with an adaptation process without retraining through pathwise updating. This enables the adaptation of solutions to new via-points with spatial and orientation constraints in real time. The algorithm encapsulates learning into prior knowledge through the use of GPs, producing an initial global policy that can incorporate position and orientation constraints. This global policy is then modified when via-points change or new constraints are added, by means of a local policy based on Pathwise Conditioning. This approach allows the algorithm to account for new task limitations while also exploiting the information previously learned from the initial model. The method has been validated both in simulation and in real-world environments, using a simulated IIWA model and a mobile manipulator robot, tested in both simulation and reality. Several real-world experiments were conducted, involving modifications of position and orientation, and highly satisfactory results were obtained in all cases. Additionally, the method was compared against other widely used approaches in the state of the art, where our approach demonstrated substantial improvements in execution time, trajectory efficiency, and the ability to generate new solutions from a single demonstration.

As future work, we propose extending the applicability of the method to dynamic systems with obstacles and to scenarios involving dual-arm manipulators, as well as incorporating guarantees of stability compliance in the adapted solutions.

ACKNOWLEDGMENT

Work supported by Advanced Mobile dual-arm manipulator for Elderly People Attendance (AMME) (PID2022-139227OB-I00) by Ministerio de Ciencia e Innovacion.

REFERENCES

[1] W. Li, Y. Hu, Y. Zhou, and D. T. Pham, "Safe human–robot collaboration for industrial settings: a survey," *Journal of Intelligent Manufacturing*, vol. 35, no. 5, pp. 2235–2261, 2024.

[2] A. Billard, S. Mirrazavi, and N. Figueroa, *Learning for adaptive and reactive robot control: a dynamical systems approach*. Mit, 2022.

[3] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *IEEE Transactions on Cybernetics*, 2024.

[4] A. Prados, A. Mora, B. López, J. Muñoz, S. Garrido, and R. Barber, "Kinesthetic learning based on fast marching square method for manipulation," *Applied Sciences*, vol. 13, no. 4, p. 2028, 2023.

[5] P. De Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *Adv. in neural info. processing*, vol. 32, 2019.

[6] J. Spencer, S. Choudhury, A. Venkatraman, B. Ziebart, and J. A. Bagnell, "Feedback in imitation learning: The three regimes of covariate shift," *arXiv preprint arXiv:2102.02872*, 2021.

[7] C. Feng, Z. Liu, W. Li, X. Lu, Y. Jing, and Y. Ma, "Improved gaussian mixture model and gaussian mixture regression for learning from demonstration based on gaussian noise scattering," *Advanced Engineering Informatics*, vol. 65, p. 103192, 2025.

[8] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using gaussian mixture models," in *6th int. conf. on Autonomous agents and multiagent systems*, 2007, pp. 1–8.

[9] C. Ye, J. Yang, and H. Ding, "Bagging for gaussian mixture regression in robot learning from demonstration," *Journal of Intelligent Manufacturing*, vol. 33, no. 3, pp. 867–879, 2022.

[10] Z. Man, L. Fengming, Q. Wei, L. Yibin, and S. Rui, "Robot bolt skill learning based on gmm-gmr," in *International Conference on Intelligent Robotics and Applications*. Springer, 2021, pp. 235–245.

[11] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics Research. The Eleventh International Symposium: With 303 Figures*. Springer, 2005, pp. 561–572.

[12] M. Tavassoli, S. Katyara, M. Pozzi, N. Deshpande, D. G. Caldwell, and D. Prattichizzo, "Learning skills from demonstrations: A trend from motion primitives to experience abstraction," *IEEE Tran. on Cognitive and Developmental Systems*, vol. 16, no. 1, pp. 57–74, 2023.

[13] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," *Adv. in neural info. processing*, vol. 26, 2013.

[14] S. Ruan, X. Meng, and G. S. Chirikjian, "Primp: Probabilistically-informed motion primitives for efficient affordance learning from demonstration," *IEEE Trans. on Rob.*, vol. 40, pp. 68–87, 2024.

[15] A. Liu, S. Zhan, Z. Jin, and W.-A. Zhang, "A variable impedance skill learning algorithm based on kernelized movement primitives," *IEEE Trans. on Industrial Electronics*, vol. 71, no. 1, pp. 870–879, 2023.

[16] Y. Huang, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *Int. Jou. of Rob. Res.*, vol. 38, no. 7, pp. 833–852, 2019.

[17] M. Schneider and W. Ertel, "Robot learning by demonstration with local gaussian process regression," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 255–260.

[18] 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.

[19] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *2016 ICRA*. IEEE, 2016, pp. 9–15.

[20] M. P. Deisenroth and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 37, no. 2, pp. 408–423, 2013.

[21] J. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. Deisenroth, "Efficiently sampling functions from gaussian process posteriors," in *Int. Conf. on ML*. PMLR, 2020, pp. 10 292–10 302.

[22] J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth, "Pathwise conditioning of gaussian processes," *Journal of Machine Learning Research*, vol. 22, no. 105, pp. 1–47, 2021.

[23] S. A. Mehta, Y. U. Ciftci, B. Ramachandran, S. Bansal, and D. P. Losey, "Stable-bc: Controlling covariate shift with stable behavior cloning," *IEEE Robotics and Automation Letters*, 2025.

[24] S. Seo, J. Lee, H. Hwang, H. Yang, and K.-E. Kim, "Mitigating covariate shift in behavioral cloning via robust stationary distribution correction," *Ad, in Neu. Inf. Proc. Sys.*, vol. 37, pp. 177–201, 2024.

[25] C. Dong and Y. Wang, "Towards generalized inverse reinforcement learning," *arXiv preprint arXiv:2402.07246*, 2024.

[26] C. Yue, T. Gao, L. Lu, T. Lin, and Y. Wu, "Probabilistic movement primitives based multi-task learning framework," *Computers & Industrial Engineering*, vol. 191, p. 110144, 2024.

[27] A. Prados, A. Mendez, G. Espinoza, N. Fernandez, and R. Barber, "f-divergence optimization for task-parameterized learning from demonstrations algorithm," in *IEEE Int. Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2024, pp. 9–14.

[28] J. Kaddour, S. Sæmundsson *et al.*, "Probabilistic active meta-learning," *Adv. in neural info. processing*, vol. 33, pp. 20 813–20 822, 2020.

[29] Y. Zhang, Z. Chen, L. Tong, H. Chen, and Y. Liu, "Hierarchical gaussian mixture based task generative model for robust meta-learning," *Adv. in neural info. processing*, vol. 36, pp. 48 662–48 685, 2023.

[30] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent service robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[31] A. Prados, S. Garrido, and R. Barber, "Learning and generalization of task-parameterized skills through few human demonstrations," *Eng. Applications of Artificial Intelligence*, vol. 133, p. 108310, 2024.

[32] A. P. Carrasco, A. M. Velasco, A. M. García, S. G. Bullón, and R. I. B. Castaño, "Learning from demonstration through synthetic data for parameterized tasks." *Jornadas de Automática*, no. 45, 2024.

[33] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *2014 IEEE ICRA*. IEEE, 2014, pp. 3339–3344.

[34] A. Mendez, A. Prados, E. Menendez, and R. Barber, "Everyday objects rearrangement in a human-like manner via robotic imagination and learning from demonstration," *IEEE Access*, vol. 12, pp. 98–119, 2024.

[35] K. Huang, X. Ji, J. Su, and X. Qu, "Task-parameterized dynamic movement primitives with reinforcement learning for improved motion planning," *IEEE Robotics and Automation Letters*, 2025.

[36] J. Li, M. Cong, D. Liu, and Y. Du, "Enhanced task parameterized dynamic movement primitives by gmm to solve manipulation tasks," *Robotic Intelligence and Automation*, vol. 43, no. 2, pp. 85–95, 2023.

[37] M. Arduengo, A. Colomé, J. Borras, and C. Torras, "Task-adaptive robot learning from demonstration with gaussian process models under replication," *IEEE RAL*, vol. 6, no. 2, pp. 966–973, 2021.

[38] J. O. von Hartz, A. Röfer, J. Boedecker, and A. Valada, "The unreasonable effectiveness of discrete-time gaussian process mixtures for robot policy learning," *arXiv preprint arXiv:2505.03296*, 2025.

[39] N. Freymuth, N. Schreiber, P. Becker, A. Taranovic, and G. Neumann, "Inferring versatile behavior from demonstrations by matching geometric descriptors," *arXiv preprint arXiv:2210.08121*, 2022.

[40] X. Yao, Y. Chen, and B. Tripp, "Improved generalization of probabilistic movement primitives for manipulation trajectories," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 287–294, 2023.

[41] T. Li and N. Figueroa, "Task generalization with stability guarantees via elastic dynamical system motion policies," in *7th Annual Conference on Robot Learning*, 2023.

[42] T. Li, S. Sun, S. S. Aditya, and N. Figueroa, "Elastic motion policy: An adaptive dynamical system for robust and efficient one-shot imitation learning," *arXiv preprint arXiv:2503.08029*, 2025.

[43] A. B. Abdessalem and K. Worden, "Automatic kernel selection for gaussian processes regression with approximate bayesian computation and sequential monte carlo," *Fron. in Built Env.*, vol. 3, p. 52, 2017.

[44] W. J. Maddox, M. Balandat, A. G. Wilson, and E. Bakshy, "Bayesian optimization with high-dimensional outputs," *Advances in neural information processing systems*, vol. 34, pp. 19 274–19 287, 2021.

[45] J. Fu, A. Liu, W.-A. Zhang, and L. Yu, "Non-parametric gaussian process movement primitive with via-point constraint for effective and safe robot skill learning," *Neurocomputing*, vol. 589, p. 127711, 2024.

[46] R. Zhang, J. Xia, J. Ma, D. Huang, X. Zhang, and Y. Li, "Human–robot interactive skill learning and correction for polishing based on dynamic time warping iterative learning control," *IEEE Transactions on Control Systems Technology*, 2024.

[47] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[48] Z. Jin, A. Liu, W.-a. Zhang, L. Yu, and C. Yang, "Gaussian process movement primitive," *Automatica*, vol. 155, p. 111120, 2023.

[49] A. Mora, A. Prados, A. Mendez, G. Espinoza, P. Gonzalez, B. Lopez, V. Muñoz, L. Moreno, S. Garrido, and R. Barber, "Adam: a robotic companion for enhanced quality of life in aging populations," *Frontiers in Neurorobotics*, vol. 18, p. 1337608, 2024.

[50] A. Prados, G. Espinoza, L. Moreno, and R. Barber, "Coordination of learned decoupled dual-arm tasks through gaussian belief propagation," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 15 917–15 924.

[51] A. Prados, G. Espinoza, A. Mendez, A. Mora, S. Garrido, and R. Barber, "Adamsim: Pybullet-based simulation environment for research on domestic mobile manipulator robots," *Jornadas de Auto.*, no. 46, 2025.