Interactive incremental learning of generalizable skills with local trajectory modulation

Markus Knauer [1,2], Alin Albu-Schäffer [1,2], Freek Stulp [1] and João Silvério [1]

Abstract—The problem of generalization in learning from demonstration (LfD) has received considerable attention over the years, particularly within the context of movement primitives, where a number of approaches have emerged. Recently, two important approaches have gained recognition. While one leverages via-points to adapt skills locally by modulating demonstrated trajectories, another relies on so-called *task-parameterized* (TP) models that encode movements with respect to different coordinate systems, using a product of probabilities for generalization. While the former are well-suited to precise, local modulations, the latter aim at generalizing over large regions of the workspace and often involve multiple objects. Addressing the quality of generalization by leveraging both approaches simultaneously has received little attention. In this work, we propose an interactive imitation learning framework that simultaneously leverages local and global modulations of trajectory distributions. Building on the kernelized movement primitives (KMP) framework, we introduce novel mechanisms for skill modulation from direct human corrective feedback. Our approach particularly exploits the concept of via-points to incrementally and interactively 1) improve the model accuracy locally, 2) add new objects to the task during execution and 3) extend the skill into regions where demonstrations were not provided. We evaluate our method on a bearing ring-loading task using a torque-controlled, 7-DoF, DLR SARA robot [1].

Index Terms—Incremental Learning, Imitation Learning, Continual Learning

I. INTRODUCTION

TASK-PARAMETERIZED Gaussian mixture models (TP-GMMs) [2] are a popular approach to encoding the variations and correlations across multiple demonstrations, facilitating skill generalization. Unlike earlier attempts such as dynamic movement primitives (DMPs) [3] and other probabilistic models for movement primitives like Gaussian mixture models (GMMs) [4], TP-GMMs are better suited for adapting to new situations, including those involving multiple objects [5]. TP-GMMs build local representations of demonstrated trajectories with respect to objects of interest, represented by their poses. They then generalize the demonstrations to new situations, by formulating generalization as a fusion problem, where each object's local model is weighed against the others
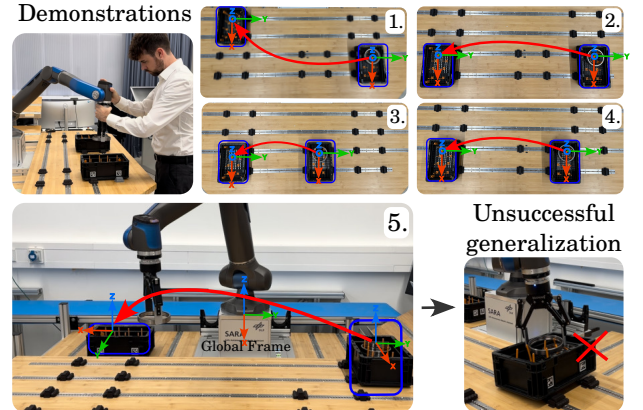
Fig. 1: By demonstrating on the same table (top row), users inadvertently introduce low variance along the vertical axis in multiple demonstrations of a ring-loading task, which prevents TP-GMM [2] from generalizing to different table heights (bottom row).

in a continuous fashion through a product of Gaussians, generating a trajectory distribution for the robot to track. Despite their adaptability, TP-GMMs are prone to modeling errors, particularly when imperfect demonstrations introduce ambiguity between objects, which affects generalization when task conditions change, see Fig. 1. In addition, introducing new objects into a learned skill (by defining new task parameters), requires providing new demonstrations and training a new model, when often simple modulations of the original model would suffice. In this paper, we propose to alleviate these issues via interactive imitation learning [6]. We do this by modulating local models from physical user feedback, both improving the generalization of taught skills – by locally correcting errors – and permitting their incremental re-use. Notably, adaptations are applied only locally, with respect to objects of interest, yielding an updated skill model which accurately generalizes new behaviors to new situations. We argue that in order to modulate skills locally, we require an underlying skill representation that 1) encodes trajectory distributions with both aleatoric and epistemic uncertainties—randomness in data and model knowledge gaps [7], and 2) allows for trajectory modulation trivially when modifications of the original demonstrations are required. To achieve this we build on the kernelized movement primitives (KMP) framework [8] (see Sec. III for a background review). Although task-parameterization of KMPs is briefly introduced in [8], the adaptation of local models has received little attention. We address the adaptation of local models by investigating *when*, *where* and *how* to add via-points. The result is an interactive learning framework of task-
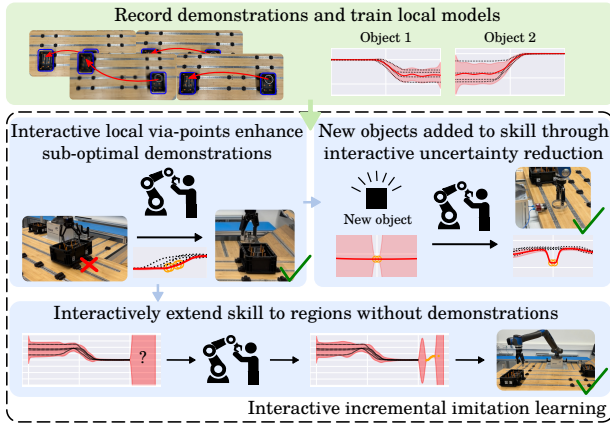
Fig. 2: Approach overview. Users provide demonstrations fitted locally to different objects using KMPs (top). The resulting model, TP-KMP, is interactively adjusted to improve generalization and augment the demonstrated skill. An interactive strategy to add via-points locally, with respect to different objects, is the cornerstone of our approach, enabling correction of skills (center-left), addition of new objects (center-right), and skill augmentation (bottom).

parameterized skills with local trajectory modulation (Sec. IV) that 1) permits the interactive correction of model imperfections locally, such that corrections 'move' with objects, 2) allows the definition of new task parameters interactively, when tasks change to require new/different objects and 3) ensures compliant interaction when extending skills beyond their initial duration by regulating stiffness based on the epistemic uncertainty, while simultaneously adding via-points. Figure 2 provides an overview of our approach. Our experimental evaluation (Sec. V) shows that our framework permits users to incrementally build on an initial model of a skill by interactively correcting errors and adding new behaviors in any task phase. We discuss the results and provide conclusions in Sec. VI–VII. An overview of the notations and acronyms used is provided in the Additional Material A–B.

II. RELATED WORK

A. Task-parameterized GMMs and variations

Several works have emerged both to address limitations of [2] and introduce new features. In [9] Huang *et al.* propose to associate confidence factors to the different task parameters, allowing to regulate their influence during task execution. Zhu *et al.* [10] introduce an algorithm to generate new data artificially by sampling from underlying models and re-training the models to improve generalization. Building on [9], Sun *et al.* [11] and Sena *et al.* [12] introduce strategies to optimize frame relevance given task objectives. Similar re-optimization is required in incremental approaches for TP models, such as [13], which also relies on an underlying GMM. Recently, Yao *et al.* [14], propose to replace the GMM representation by a ProMP [15], introducing the idea of improving generalization by modulating TP models with via-points. However, via-points are defined globally and projected locally on all frames. This creates ambiguity when task parameters change, since all frames have high confidence on the via-point positions, requiring the re-definition of via-points every time a task

changes. Although these approaches improve generalization in scenarios with non-moving objects, they do not permit adding new task parameters to existing models and, most importantly, they do not provide interactive capabilities to incrementally refine skills.

B. Interactive imitation learning

The emergence of interactive imitation learning [6] highlights the importance of complementing the strengths of classical machine learning methods with interactive mechanisms when acquiring skills in the real world. Along these lines, Franzese *et al.* [16] introduce the concept of interactive task-frame selection, focusing on object-centered skills in a similar spirit to TP approaches, despite not contemplating the learning of trajectory distributions. Interactive incremental learning approaches in LfD include state-based representations [17]–[21], where skills are learned as time-independent autonomous systems and modulated directly in task space by introducing new state-action pairs. Huang *et al.* [8] introduced KMPs, aiming for a probabilistic representation that can be adapted to pass through new via-points after the initial demonstration phase. Despite introducing a TP formulation of KMP, the authors do not explore the combination of interactive via-point-based modulation with object-centered representations.

C. Our approach and contribution

We extend on the combination of [2] with KMP-based local trajectory representations, referred to as TP-KMP, leveraging the intuitive addition of via-points after initial demonstration by specifying their location and covariance, as proposed in [8]. In previous work [14], via-points were added globally for obstacle avoidance, necessitating re-demonstration when objects were moved. In contrast, we add via-points locally, allowing them to move with the objects. Defining via-points with low covariance increases the importance of the corresponding objects in local models, enhancing the relevance of these via-points in new scenarios. Our framework uses this property, incorporating human feedback - often investigated for trajectory modulation [22] - to specify via-point locations. Building on this insight, we further develop an interactive learning framework for defining new task parameters and improving skills incrementally, by leveraging the capabilities of KMPs to encode both aleatoric and epistemic uncertainties [7], [23]. Uncertainty quantifications, important in robotics [24]–[26], allow the definition of robot behaviors through uncertainty-aware stiffness regulation strategies [27], [28]. In comparison to other approaches like [29], we rely on both epistemic and aleatoric uncertainties in a principled manner (Sec. IV-D).

In the domain of incremental learning, approaches like [30], [31], despite allowing for corrections, are either not task-parameterized or require large amounts of training data, making them less intuitive for novice users. ILoSA [20] or its extended version [32] provide a way to apply partial correction demonstrations locally. However, due to the missing aleatoric uncertainty a combination of local models is only possible using a heuristics instead of an uncertainty-based

TABLE I: Comparison of related task-parameterized approaches. Our approach models both uncertainties and offers multiple possibilities to adapt motion primitives.

	Aleatoric uncertainty	Epistemic uncertainty	Global via-point	Local via-point	New frames	Interactive modulation
TP-GMM/GMR [2]	✓	-	-	-	-	-
TP-ProMP [14]	✓	-	✓	-	-	-
vanilla TP-KMP [8]	✓*	✓*	✓	-	-	-
TP-ILoSA [32]	-	✓	✓**	✓**	-	✓
Our approach	✓	✓	✓	✓	✓	✓

* available jointly, not used for interaction
** Via-points specified simply as new datapoints through corrections

fusion like the one used in TP-KMP. In addition, using via-points instead of correction demonstrations is more efficient since the correction only has to be given once.

Finally, compared to DMPs [3], which offer spatial and temporal invariance for movement scaling, our approach additionally provides a greater flexibility by a task-parameterized formulation [2]. This enables continuous adaptation to multiple objects and purposeful interactions throughout the task, surpassing the fixed goal attractor limitations of DMPs.

In Table I we compare our approach to other task-parameterized approaches on feature level. As shown in [5] task-parameterized approaches are more suitable for adapting to new situations than non-task-parameterized approaches like [30] or [31], including those involving multiple objects. To the best of our knowledge, we present the first task-parameterized approach allowing skill extensions with new frames and the application of local via-points, making it possible to move corrections with their corresponding objects and allowing interactive modulations, where corrections/extensions can be provided and incorporated into the policy online during execution.

III. PRELIMINARIES

Let us denote a set of demonstrations by $\{\{s_{h,m}, \xi_{h,m}\}_{h=1}^{H}\}_{m=1}^{M}$, where $s \in \mathbb{R}^{\mathcal{I}}$ and $\xi \in \mathbb{R}^{\mathcal{O}}$ represent input and output, and $\mathcal{I}$, $\mathcal{O}$, M, H, are the dimensions of input and output, number of demonstrations and trajectory length, respectively. Similarly to many popular LfD approaches [4], [8], [15], we focus on extracting the relationship between s and ξ from demonstrations. Depending on the task, s is often time or robot state while ξ represents desired poses or velocities. In Sec. III and Sec. IV we keep these generic except when concrete examples help explain new concepts.

A. Task parameterized movement models

In TP models [2] a *frame* $p = 1, \ldots, P$ is described by so-called *task parameters* $b^{(p)}, A^{(p)}$, which represent the position and orientation of an object with respect to a common reference frame (e.g. the robot base)[1], where demonstrations are recorded. Demonstrated outputs are projected locally, into the coordinate systems of different objects, through $\xi^{(p)} = A^{(p)^{-1}} (\xi - b^{(p)})$. Local datasets are modeled probabilistically yielding a Gaussian distribution $\mathcal{N}(\mu^{(p)}, \Sigma^{(p)})$ for

---

[1]Task parameters can represent a number of affine transformations commonly found in robotics [2]. Here, as in other works [9]–[11], [14], we use them to represent the coordinate systems of objects of interest.

---

every input s. During skill execution, local distributions are mapped to the common frame, as both task parameters and inputs change with time t, through $\hat{\mu}_t^{(p)} = A_t^{(p)} \mu_t^{(p)} + b_t^{(p)}$, $\hat{\Sigma}_t^{(p)} = A_t^{(p)} \Sigma_t^{(p)} A_t^{(p)^{\top}}$. Values of $b^{(p)}, A^{(p)}$ may differ from those seen during the demonstrations, and, through $\hat{\mu}_t^{(p)}$, $\hat{\Sigma}_t^{(p)}$, each frame provides a model of ξ from its perspective. The distributions from different coordinate systems are fused by a product of Gaussians, resulting in a new distribution $\mathcal{N}(\mu_t, \Sigma_t)$, in the common frame, with parameters

$$\mu_t = \Sigma_t \sum_{p=1}^{P} \hat{\Sigma}_t^{(p)^{-1}} \hat{\mu}_t^{(p)}, \quad \Sigma_t = \left(\sum_{p=1}^{P} \hat{\Sigma}_t^{(p)^{-1}} \right)^{-1}. \quad (1)$$

The solution of (1) favors models with low variance, being an efficient way to extract features from local models that were consistent across demonstrations, facilitating generalization to new situations.

B. Kernelized movement primitives (KMPs)

KMPs [8] are used in LfD to predict the distribution of ξ given observations of s. A KMP is initialized with a *reference trajectory distribution* comprised of N Gaussians with parameters $\{\mu_n, \Sigma_n\}_{n=1}^{N}$, computed from human demonstrations for inputs $s_{n=1,\ldots,N}$ using a GMM. For a test input s^*, the expectation and covariance of $\xi(s^*)$ are given by

$$\mathbb{E}[\xi(s^*)] = k^* (K + \lambda_1 \Sigma)^{-1} \mu, \quad (2)$$

$$\text{cov}[\xi(s^*)] = \alpha \left(k^{**} - k^* (K + \lambda_2 \Sigma)^{-1} k^{*\top} \right), \quad (3)$$

where $K = [\hat{k}(s_1)^{\top}, \ldots, \hat{k}(s_N)^{\top}]$, $k^* = \hat{k}(s^*)$, with $\hat{k}(s_i) = [k(s_i, s_1), \ldots, k(s_i, s_N)]$, $k^{**} = k(s^*, s^*)$, $k(s_i, s_j) = k(s_i, s_j)I$, where I is an identity matrix, and $k(s_i, s_j)$ is a kernel function. Moreover, $\mu = [\mu_1^{\top} \ldots \mu_N^{\top}]^{\top}$, $\Sigma = \text{blockdiag}(\Sigma_1, \ldots, \Sigma_N)$ and λ_1, λ_2, α are hyperparameters. The kernel matrices are denoted as K, k^* and k^{**}. From (2)–(3) it follows that if, for a certain μ_n, the covariance Σ_n is small, the expectation at s_n will be close to μ_n. This provides a principled way for trajectory modulation. Indeed, if, for a new input $\bar{s}$, one wants to ensure that the expectation passes through a desired $\bar{\mu}$, it suffices to manually add the pair $\{\bar{\mu}, \bar{\Sigma}\}$ to the reference distribution provided that $\bar{\Sigma}$ is small enough. This both makes (2) closely match $\bar{\mu}$ and lowers the covariance (3) to match $\bar{\Sigma}$.

IV. INTERACTIVE LOCAL TRAJECTORY MODULATION WITH TP-KMP

Similarly to [8], we define a local KMP as a model $\Theta^{(p)} = \{b^{(p)}, A^{(p)}, D^{(p)}\}$ with associated task parameters $b^{(p)}, A^{(p)}$ and $D^{(p)} = \{s_n^{(p)}, \mu_n^{(p)}, \Sigma_n^{(p)}\}_{n=1}^{N}$. $\mu^{(p)}, \Sigma^{(p)}$ are computed from output data projected locally $\xi^{(p)}$, using a GMM. A TP-KMP is a set of P local KMPs: $\Theta = \{\Theta^{(p)}\}_{p=1}^{P}$, where each local KMP generates a distribution $\mathcal{N}(\mu^{(p)}, \Sigma^{(p)})$, computed from (2)–(3), which is used in (1). We introduce an approach to interactively add via-points to local KMPs at any moment of a task, allowing users to intuitively improve models trained on sub-optimal demonstrations.

A. Interactive trajectory modulation with local via-points

In our approach, users add via-points via physical corrections locally, in different object frames, as opposed to in a common global frame. This enables the adaptation of robot behavior without retraining the model from scratch. Adding via-points in the relevant local frames has the advantage that corrections 'move' with the objects when task conditions change, facilitating generalization. The via-point mechanism of KMP entails the definition of a small covariance matrix, automatically assigning high importance in the Gaussian product (1) to local KMPs that receive new via-points.

Algorithm 1 *Trajectory modulation with local via-points*

1: **Define** external force threshold γ_F, distance threshold γ_ξ and via-point variance γ_Σ

2: **Input**: P local KMPs $\Theta^{(p)} = \{\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}, \boldsymbol{D}^{(p)}\}_{p=1}^{P}$ trained from $\{\{\boldsymbol{s}_{h,m}, \boldsymbol{\xi}_{h,m}\}_{h=1}^{H}\}_{m=1}^{M}$

3: **for** time-step t in all time-steps T **do**

4:     **if** interaction is triggered (see IV-A) **then**
    ▷ Create global via-point

5:         $\bar{v}_t = \{\bar{s} = t, \bar{\boldsymbol{\mu}} = \boldsymbol{\xi}_t, \bar{\boldsymbol{\Sigma}} = \gamma_\Sigma \boldsymbol{I}\}$
    ▷ Find p^* (see IV-A)

6:         $p^* = \operatorname{argmin}_p ||\bar{\boldsymbol{\mu}} - \boldsymbol{b}^{(p)}||$
    ▷ Map via-point $\bar{v}_t$ locally to p^* (see III-A)

7:         $\bar{v}^{(p^*)} = \{\bar{s}^{(p^*)}, \bar{\boldsymbol{\mu}}^{(p^*)}, \bar{\boldsymbol{\Sigma}}^{(p^*)}\}$
    ▷ Update local KMP p^* with via-point

8:         $\boldsymbol{D}^{(p^*)} \leftarrow \boldsymbol{D}^{(p^*)} \cup \bar{v}^{(p^*)}$

9: **Recompute** $\boldsymbol{K}^{(p)}, \boldsymbol{k}^*_{(p)}$ and $\boldsymbol{k}^{**}_{(p)}$ for all p where via-points were added (see III-B)

B. Defining and adding local via-points

To incorporate kinesthetic feedback from users during task execution, we introduce via-points at specific inputs and outputs, where corrections are made. Let us assume a time-driven TP-KMP, where $\boldsymbol{s}^{(p)} = t$, and $\boldsymbol{\xi}^{(p)} = \boldsymbol{x}^{(p)}$ is the end-effector position mapped to frame p. We further assume that the robot tracks a reference $\hat{\boldsymbol{\xi}}_t = \boldsymbol{\mu}_t$ computed from (1) with a stiffness that is low enough to allow deviations from the desired path through physical interaction. We propose three different feedback modalities to trigger new via-points:

- **Distance:** via-points are added if the distance between the measured output $\boldsymbol{\xi}_t$ deviates from the reference $\hat{\boldsymbol{\xi}}_t$ by a pre-defined threshold $||\hat{\boldsymbol{\xi}}_t - \boldsymbol{\xi}_t|| > \gamma_\xi$.
- **Force:** via-points are added if the external force applied at the end-effector $\boldsymbol{F}_t$ exceeds a pre-defined threshold $||\boldsymbol{F}_t|| > \gamma_F$.
- **Button press:** via-points can be triggered through a Boolean-type input such as pressing a button, e.g. on a keyboard or on the robot's button interfaces when available.

In all cases, a via-point is defined globally, in a common frame, as $\bar{v}_t = \{\bar{s} = t, \bar{\boldsymbol{\mu}} = \boldsymbol{\xi}_t, \bar{\boldsymbol{\Sigma}} = \gamma_\Sigma \boldsymbol{I}\}$, where γ_Σ is a small scalar factor. Different feedback modalities provide users with various options depending on the robot and task. Force-based adaptation is well-suited for tasks (or sub-tasks) that require little or no contact with the environment, where the robot can interpret external forces as human intention. A distance criteria is suitable for tasks where the robot has low stiffness and thus would not measure high external forces when perturbed. In Sec. V, we showcase the role of various feedback modalities. When a new global via-point $\bar{v}_t$ is defined, we rely on a proximity-based criteria to determine which KMP to apply it to. For this, we identify the closest frame to the via-point, $p^* = \operatorname{argmin}_p ||\bar{\boldsymbol{\mu}}_t - \boldsymbol{b}^{(p)}||$. The via-point is then mapped to the selected frame through $\bar{\boldsymbol{\mu}}^{(p^*)} = (\boldsymbol{A}^{(p^*)})^{-1}(\bar{\boldsymbol{\mu}} - \boldsymbol{b}^{(p^*)})$, $\bar{\boldsymbol{\Sigma}}^{(p^*)} = (\boldsymbol{A}^{(p^*)})^{-1}\bar{\boldsymbol{\Sigma}}(\boldsymbol{A}^{(p^*)\top})^{-1}$ (see Sec. III for details) and added to its KMP, entailing the recomputation of kernel-related matrices $\boldsymbol{K}_{(p^*)}, \boldsymbol{k}^*_{(p^*)}$ and $\boldsymbol{k}^{**}_{(p^*)}$. The algorithm for interactively adding via-points to local frames is summarized in Algorithm 1.

C. Adding new objects through interactive via-point definition

Our approach permits adding new objects to a skill during runtime, without requiring a new set of demonstrations. Instead, we leverage Algorithm 1 to build on an existing TP-KMP, $\Theta = \{\Theta^{(p)}\}_{p=1}^{P}$, by interactively adding via-points in the frames of new objects. To achieve this, we employ two key steps. Firstly, we associate to the new task parameters $\boldsymbol{b}^{(P+1)}, \boldsymbol{A}^{(P+1)}$ a *placeholder* local KMP with the same inputs as other local KMPs, zero means and high variances, defined by a large scalar γ_D. Due to the high variance, the placeholder KMP has a negligible influence in the Gaussian product (1), not affecting the task unless via-points are added to it. Next, we use Algorithm 1 to add via-points to the placeholder KMP, interactively reducing the uncertainty at precise locations, enabling the model to adapt without requiring new demonstrations. Algorithm 2 summarizes the procedure to add new objects.

Algorithm 2 *Interactively adding a new object to a TP-KMP*

1: **Input:** TP-KMP $\Theta = \{\Theta^{(p)}\}_{p=1}^{P}$, variance prior γ_D, new task parameters $\boldsymbol{b}^{(P+1)}, \boldsymbol{A}^{(P+1)}$.

2: **Create a *placeholder* local KMP**
  - Define $\boldsymbol{D}^{(P+1)} = \{\boldsymbol{s}_N^{(P+1)} = \boldsymbol{s}_n^{(P)}, \boldsymbol{\mu}_n^{(P+1)} = \boldsymbol{0}, \boldsymbol{\Sigma}_n^{(P+1)} = \gamma_D \boldsymbol{I}\}_{n=1}^{N}$
  - Define local KMP: $\Theta^{(P+1)} = \{\boldsymbol{b}^{(P+1)}, \boldsymbol{A}^{(P+1)}, \boldsymbol{D}^{(P+1)}\}$
  - Compute $\boldsymbol{K}_{(P+1)}, \boldsymbol{k}^*_{(P+1)}$ and $\boldsymbol{k}^{**}_{(P+1)}$
  - Add new local KMP to TP-KMP: $\Theta \leftarrow \Theta \cup \Theta^{(P+1)}$, $P = P + 1$

3: **Add via-points to new KMP interactively using Algo.1**

D. Uncertainty-aware skill extension in regions without demonstrations

Popular variable impedance schemes found in LfD regulate the robot stiffness by the inverse of covariance matrices [23], [28]. When the latter represent the aleatoric uncertainty, they provide an efficient way for robots to be more precise, by being stiffer, where demonstrations showed less variance, following a *minimum intervention control* principle [28], [33]. In the case of epistemic uncertainty, such schemes contribute to better

compliance when the robot is uncertain about its actions [23]. In kernel-based methods, kernel hyperparameters are typically optimized for the training data, but their choice influences the behavior of the model in regions where data was not shown. For instance, the kernel length depends on the scale of the input domain, but it also dictates how quickly the epistemic uncertainty increases when moving away from the training data.[2] Having ways to clearly distinguish between the two types of uncertainty enables users to better design variable impedance strategies. Similarly to [34], the covariance prediction of KMP can be decomposed into two terms, corresponding to aleatoric and epistemic components (see Appendix 1 for the derivation):

$$\text{cov}\left[\boldsymbol{\xi}(\boldsymbol{s}^*)\right] =$$
$$\underbrace{\boldsymbol{k}^{**} - \boldsymbol{k}^* \boldsymbol{K}^{-1} \boldsymbol{k}^{*\top}}_{\boldsymbol{\Sigma}_{\text{ep}}^*} + \underbrace{\boldsymbol{k}^* \left(\boldsymbol{K} + \boldsymbol{K}(\lambda_2 \boldsymbol{\Sigma})^{-1} \boldsymbol{K}\right)^{-1} \boldsymbol{k}^{*\top}}_{\boldsymbol{\Sigma}_{\text{al}}^*} \quad (4)$$

The term $\boldsymbol{\Sigma}_{\text{ep}}^*$ is the same as the variance prediction in Gaussian process regression [35], corresponding to the epistemic part of the KMP covariance. The remaining term $\boldsymbol{\Sigma}_{\text{al}}^*$ gives the aleatoric uncertainty. Using a Cartesian impedance controller, from (4), we propose to compute the robot end-effector stiffness[3] using

$$\boldsymbol{G}_P = w_1 \cdot \left(\delta_{\text{ep}} \boldsymbol{\Sigma}_{\text{ep}}^*\right)^{-1} + w_2 \cdot \left(\delta_{\text{al}} \boldsymbol{\Sigma}_{\text{al}}^*\right)^{-1}, \quad (5)$$

where $w_1 \in [0, 1]$ is a sigmoid function that depends on the epistemic uncertainty $w_1(\sigma_{\text{ep}}^2) = 1/(1 + e^{-c_1(\sigma_{\text{ep}}^2 - c_2)})$, with $\sigma_{\text{ep}}^2 = \text{tr}(\boldsymbol{\Sigma}_{\text{ep}}^*)/\mathcal{O}$, and $w_2 = 1 - w_1$. Through the parameters $c_1 > 0, c_2$ one is able to regulate the rate at which the robot stiffness switches between being governed by $\boldsymbol{\Sigma}_{\text{ep}}^*$ and $\boldsymbol{\Sigma}_{\text{al}}^*$, while ensuring continuity in the resulting accelerations. Parameters $\delta_{\text{ep}}, \delta_{\text{al}}$ re-scale the uncertainties without modifying the kernel parameters, allowing, for example, $\boldsymbol{\Sigma}_{\text{ep}}^*$ to have a stronger influence in $\boldsymbol{G}_P$ far from the training data, which permits a faster increase in robot compliance. We leverage (5) to facilitate the acquisition of new data outside of the training region through physical interactions, while being optimal in a minimal intervention sense [28], [33], in regions where data was provided.

V. EVALUATION

A. Evaluation on real robot

We evaluate our approach on a torque-controlled 7-DoF robot in an industrial scenario where an inner ring of a ball bearing needs to be transferred between two boxes ('box 1' and 'box 2') placed at different locations on a workbench.[4] We provide $M = 4$ demonstrations with different box positions, see Fig. 1-top. We use a time-driven representation with

---

[2]Similar arguments can be made for other hyperparameters such as the noise variance, see [23].

[3]We assume a controller $\boldsymbol{u} = \boldsymbol{G}_P(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}) - \boldsymbol{G}_D\dot{\boldsymbol{\xi}}$, where $\boldsymbol{G}_P, \boldsymbol{G}_D$ are stiffness and damping gains, $\hat{\boldsymbol{\xi}}, \boldsymbol{\xi}, \dot{\boldsymbol{\xi}}$ are the end-effector desired state, current state and current velocity. $\boldsymbol{G}_D$ changes with the joint configuration, to ensure that the system is critically damped in Cartesian space [36]. Redundancy is further addressed at joint level using a null space term that keeps the elbow in a natural position.

[4]You can find more information and videos at github.com/DLR-RM/interactive-incremental-learning

$s_{h,m} = t_{h,m}/T_m$, where t is a time step, and learn the end-effector position $\boldsymbol{\xi}_{h,m} = \boldsymbol{x}_{h,m}$. In order to easily re-scale the skill duration, we map all the inputs to the interval $[0, 1]$ by dividing them by the duration of each demonstration T_m. The experiments start with $P = 2$, with task parameters $\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}$ representing the box positions and orientations, respectively. All local KMPs were initialized from GMMs with 12 components, trained on locally projected data, and $N = 500$ inputs, equally spread over the input space. We chose a Matérn kernel ($\nu = 5/2$) with length scale $l = 0.1$ and noise variance 1.0 (see [35]). Other KMP hyperparameters were $\lambda_1 = 0.1$, $\lambda_2 = 1$, $\alpha = 1$, chosen empirically. For completeness, we provide an overview of hyperparameters in Appendix 2. In all experiments, via-points are added with $\gamma_\Sigma = 1 \times 10^{-8}$. Our algorithm successfully generalizes to novel box positions. However, when they deviate significantly from the demonstrated ones, TP-GMM [2], as well as the original TP-KMP formulation [8] fail (see Fig. 1-bottom).

B. Experiment 1: Improving generalization with interactively-defined local via-points

Since all demonstrations are given at the same height, the model has a high confidence for the z coordinate in all frames, which leads to poor generalization when one of the boxes is moved to a different height (Fig. 1-bottom). If box 2 is moved to a lower height, the trade-off found by the model, from the expected vertical motions in the two frames, is not high enough to successfully move the ring out of box 1, leading to a collision. Using Algorithm 1, we have the ability to directly correct the robot and set via-points based on corrections made online. In this experiment we used an external force trigger, with threshold $\gamma_F = 20N$. Figure 3 shows the results obtained for this experiment. With the robot running the model, a user applies a force at the robot end-effector, helping it avoid the collision and successfully move the ring out of the box. Via-points are added to the frame of box 1 and, through $\boldsymbol{b}^{(1)}, \boldsymbol{A}^{(1)}$, are mapped to the global frame. With the added via-points, the robot achieves a success rate of 93% across 15 box configurations (with three different heights), improved over the 60% success rate of TP-GMM [2] and original TP-KMP [8] baselines (see Additional Material C).

C. Experiment 2: Interactively adding a new object to skill

We introduce a new challenge by requiring the robot to inspect the ring using a camera, before placing it in box 2. For this, the user increments the existing skill by interactively adding via-points to a new frame, given by the camera pose, using Algorithm 2. The new via-points adapt the robot's trajectory to pass in front of the camera before placing the ring in box 2. In this experiment we use the button interfaces in the robot's last link to trigger new via-points through button presses. We further use $P = 3$, due to the new frame, and $\gamma_D = 1 \times 10^4$. The experimental results are shown in Fig. 4. With the task running, the user adds new via-points near the camera (top-right), reducing the uncertainty in the camera frame. Once the model is updated with the new via-points, new executions pass in front of the camera (bottom-right).
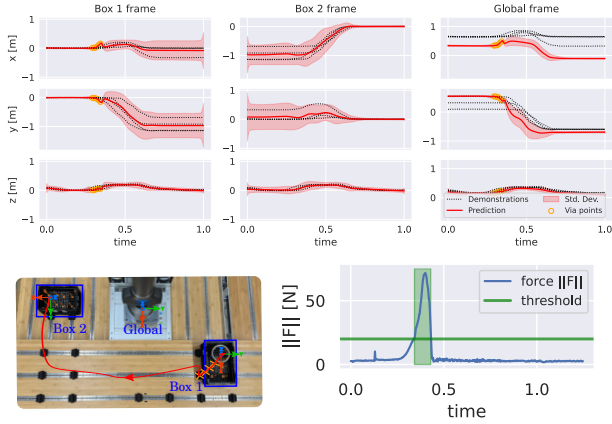
Fig. 3: Improving generalization with interactively-defined local via-points. Since the robot would fail, a correction is shown close to box 1, indicated by via-points in orange. The threshold for the external force γ_F is set to 20 N.
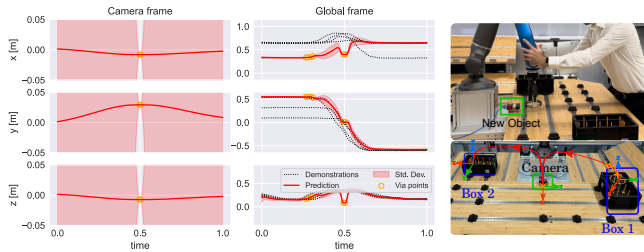

Fig. 4: Interactively adding a new object to existing skill. We add a camera to the scene, associate it with a new frame and add via-points to pass in front of the camera, this time by pressing a button.

D. Experiment 3: Incremental learning at non-demonstrated inputs

We further add a camera to the robot end-effector, which validates the correct insertion of the ring in box 2. This requires the robot to move up after placing the ring – a skill extension to a set of inputs that were not shown in the demonstrations ($t > 1.0$). We leverage (5) to ensure that the robot becomes compliant by swiftly lowering its stiffness in response to the increase in epistemic uncertainty.

We set $c_1 = 5 \times 10^3$, $c_2 = 1.5 \times 10^{-3}$, $\delta_{ep} = 1 \times 10^3$, $\delta_{al} = 1$, chosen empirically. We consider a baseline where the gains are computed from the original KMP covariance (3) as $\boldsymbol{G}_P = (\text{cov}\left[\boldsymbol{\xi}(\boldsymbol{s}^*)\right])^{-1}$. To address numerical issues and ensure a maximum stiffness value we regularize $\text{cov}\left[\boldsymbol{\xi}(\boldsymbol{s}^*)\right]$, $\delta_{al}\boldsymbol{\Sigma}_{al}^*$, $\delta_{ep}\boldsymbol{\Sigma}_{ep}^*$ with a small scalar 1.5×10^{-3}. Figure 5 shows that using our approach (5) the interaction forces after $t = 1.0$ are negligible. Note that far from the training data, the KMP expectation (2) converges to zero, as we assume a zero mean prior (similarly to GPs [35]). With the improved compliance behaviors introduced by (5), the user is able to add via-points such that the robot moves up after the ring insertion. In this experiment we used a distance threshold with $\gamma_\xi = 0.2m$, taking advantage of the reference trajectory going to zero to trigger the definition of via-points. The bottom-left image in Fig. 6 shows the increase in distance as via-points are added, while the plots on the right show the resulting robot motion and uncertainties. Appendix 3 shows the detailed stiffness
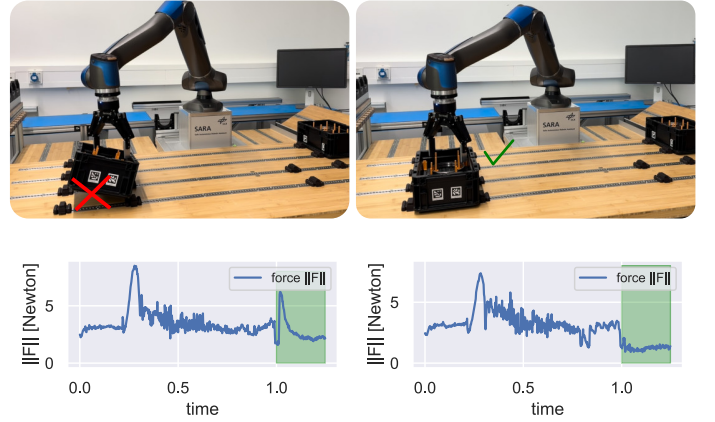

Fig. 5: **Left:** Computing stiffness gains from (3) leads to increased interaction forces at non-demonstrated inputs. **Right:** Using our approach (4)–(5) the robot reacts quickly to the increase in the epistemic uncertainty, lowering its stiffness. $||\mathbf{F}||$ represents the Euclidean norm of vector $\mathbf{F}$.

TABLE II: Mean $\pm$ standard deviation distance from ground truth for the pick-and-place task from Yao *et al.* [14]. All results besides TP-KMP [8] and our approach are generated using the data and code provided by [14]. We added via-points to the start and stop positions.

	Start (mm)	End (mm)	Average (mm)
ProMP [15]	107.78 ± 7.99	137.84 ± 18.10	250.68 ± 137.13
TP-GMM (time-based) [2]	28.22 ± 14.90	74.11 ± 48.06	58.76 ± 21.43
TP-GMM (dynamic) [2]	27.54 ± 15.56	129.42 ± 100.18	83.45 ± 43.01
KMP [8]	25.80 ± 11.39	73.99 ± 36.44	57.48 ± 21.62
TP-ProMP [14]	32.62 ± 11.39	48.60 ± 20.68	$\mathbf{48.51 \pm 12.21}$
TP-KMP [8]	39.30 ± 29.06	6.93 ± 5.08	51.95 ± 26.44
Our approach (with via-points)	$\mathbf{3.88 \times 10^{-4}}$ $\mathbf{\pm 2.54 \times 10^{-4}}$	$\mathbf{1.16 \times 10^{-3}}$ $\mathbf{\pm 6.38 \times 10^{-4}}$	50.23 ± 25.58

profiles from Experiment 3 including a sensitivity analysis of the kernel length scales.

E. Evaluation on toy example

In Table II we compare the performance of our approach in IV-A to other LfD approaches (with a strong focus on task-parameterized formulations). For this comparison, we follow the evaluation framework of Yao et al. [14] and ran both TP-KMP [8] and our interactive approach on the pick-and-place task; the results are based on 100 runs with different start and end points. Our approach showed significant improvement at both the start and end points, delivering the best results in this comparison for start and end precision. This is due to showed corrections at the start and end-point, which we did by locally adding via-points at the origin of the corresponding reference frames. Our approach also compares favorably in terms of average precision, despite not including other via-points than at start and end, similarly to [14].

VI. DISCUSSION

A. Analysis of the results

Fig. 3 shows that in Experiment 1 an external force trigger successfully allows for the definition of via-points in the nearest frame (that of box 1). Thanks to a small via-point variance, the via-points are mapped to the global frame by (1) improving generalization quality by avoiding a collision. Fig. 4, Experiment 2 illustrates how the definition of a placeholder frame
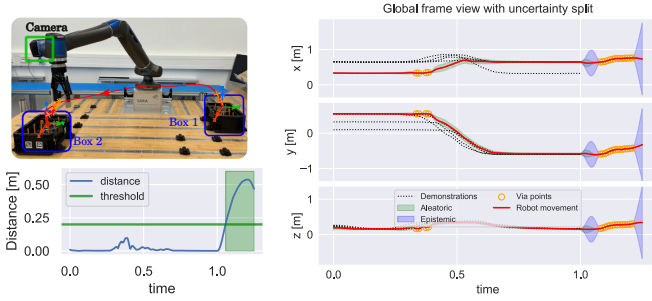
Fig. 6: Adding via-points at $t > 1.0$, triggered by distance threshold $\gamma_\xi = 20m$ leads to increase of uncertainty outside of demonstrated area.

TABLE III: Hyper-parameter overview

Hyper-parameter	Used in paper	Validated range/options	Remark
Nr. of Gaussians to initialize GMM	12	[7; 40]	$\uparrow \mathbb{N}$: for more complex/longer tasks
Regularization term KMP mean λ_1	0.1	[0.1; 1.0]	$\uparrow \lambda_1$: closer following the mean distribution of the KMP
Regularization term KMP variance λ_2	1.0	[0.1; 1.0]	$\uparrow \lambda_2$: closer following the variance distribution of the KMP
Scaling factor for KMP covariance α	1.0	[0.1; 1.0]	-
Length scale of kernel l	0.1	[0.1; 0.7]	$\uparrow l$: for smoother predictions, but less accurate
Kernel function $k(.,.)$	matern2	matern2 / rbf	-
Sigmoid function parameter c_1	5×10^3	[1; 1×10^4]	$\uparrow c_1$: control gains switch more quickly between aleatoric and epistemic uncertainty
Sigmoid function parameter c_2	1.5×10^{-3}	[1×10^{-4}; 1]	$\uparrow c_2$: epistemic uncertainty takes over for larger values of σ_{ep}^2
Epistemic scaling factor δ_{ep}	1×10^3	[1; 1×10^4]	$\uparrow \delta_{ep}$: increases the influence of epistemic uncertainty
Aleatoric scaling factor δ_{al}	1	[1; 1×10^4]	$\uparrow \delta_{al}$: increases the influence of aleatoric uncertainty

with large variance can be used in combination with the via-point insertion mechanism from Experiment 1 to introduce behaviors with respect to objects that were not present in the demonstrations. Particularly, one observes that the variance in frame 3 decreases after the via-points are added, with the latter being successfully mapped to the global frame. Finally, Experiment 3 shows how our stiffness regulation approach leverages the epistemic uncertainty to enhance the robot's compliance beyond the initial set of demonstrations. While one could argue that a similar effect could be achieved by manually lowering the stiffness at $t > 1.0$, this would require manually keeping track of the duration of demonstrations, as well as the exact locations of newly added via-points. Our approach automates this by relying on the data properties, increasing the epistemic uncertainty both before and after via-points (Fig. 6-right).

B. Limitations

In our approach, via-points are added in all Cartesian DoFs, even though a correction might only be required in a subset thereof (*e.g.* height in the first experiment). Since KMPs allow the definition of via-point covariances with different diagonal entries, one can, in principle, selectively set higher precision only on the DoFs that receive a corrective action (keeping the others as in the training data). This, however, requires a more complex interaction mechanism which extracts the user intention, a topic that we plan to address in future research. Another possible limitation is that our approach currently does not allow for combining new via-points with existing ones at the same location. If a via-point is added at a location where one already existed, the only way to account for the newly added information is to replace the pre-existing via-point. To provide users with more options for skill re-use, in future work we will investigate mechanisms to interactively remove via-points.

VII. CONCLUSION

We presented an interactive imitation learning framework that leverages both local and global modulations of trajectory distributions to address the problem of generalization in LfD. To improve the generalization quality and incrementally add new features to a demonstrated skill, the framework allows the interactive definition of local via-points. This is facilitated by a variable impedance scheme that leverages epistemic uncertainties to augment skills beyond the demonstrations. Our results, evaluated on a ring-loading task using a torque-controlled, 7-DoF robot, show that our framework permits users to incrementally build on an initial model of a skill by interactively correcting errors and adding new behaviors in any phase of the task. This work has significant implications for the development of robots that can learn from demonstration and generalize their skills to new situations, making them more versatile and effective in real-world applications.

APPENDIX

1) Decomposition of KMP covariance into a sum of epistemic and aleatoric terms: We here show that, similarly to [34], the covariance prediction of KMP can be decomposed into two distinct terms, corresponding to aleatoric and epistemic components. Using the Woodbury identity $(A + UBV)^{-1} = A^{-1} - A^{-1}U\left(B^{-1} + VA^{-1}U\right)^{-1} VA^{-1}$ with $V = U = I$, we can re-write (3) as:

$$\mathrm{cov}\left[\boldsymbol{\xi}(\boldsymbol{s}^*)\right] =$$
$$= \boldsymbol{k}^{**} - \boldsymbol{k}^*\left[\boldsymbol{K}^{-1} - \boldsymbol{K}^{-1}\left(\boldsymbol{K}^{-1} + (\lambda_2\boldsymbol{\Sigma})^{-1}\right)^{-1}\boldsymbol{K}^{-1}\right]\boldsymbol{k}^{*\top}$$
$$= \boldsymbol{k}^{**} - \boldsymbol{k}^*\boldsymbol{K}^{-1}\boldsymbol{k}^{*\top} + \boldsymbol{k}^*\boldsymbol{K}^{-1}\left(\boldsymbol{K}^{-1} + (\lambda_2\boldsymbol{\Sigma})^{-1}\right)^{-1}\boldsymbol{K}^{-1}\boldsymbol{k}^{*\top}$$
$$= \underbrace{\boldsymbol{k}^{**} - \boldsymbol{k}^*\boldsymbol{K}^{-1}\boldsymbol{k}^{*\top}}_{\boldsymbol{\Sigma}_{ep}^*} + \underbrace{\boldsymbol{k}^*\left(\boldsymbol{K} + \boldsymbol{K}(\lambda_2\boldsymbol{\Sigma})^{-1}\boldsymbol{K}\right)^{-1}\boldsymbol{k}^{*\top}}_{\boldsymbol{\Sigma}_{al}^*},$$

where, in the last step, we used $A^{-1}\left(A^{-1} + B^{-1}\right)^{-1} A^{-1} = \left(A + AB^{-1}A\right)^{-1}$ (omitting α for the sake of the derivation). Note the clear separation between the two terms in (4). The term $\boldsymbol{\Sigma}_{ep}^*$ is the same as the variance prediction in Gaussian process regression [35], corresponding to the epistemic part of the KMP covariance. The remaining term $\boldsymbol{\Sigma}_{al}^*$ gives the aleatoric uncertainty.

2) Hyper-parameter overview: Table III shows the different hyperparameters used in this paper and a tested range of values to provide a guideline for choosing hyperparameters.

3) Detailed stiffness profiles for Experiment 3 with kernel length scale sensitivity analysis: Figures 7–8 give a detailed view of the stiffness profiles which we observed in Experiment 3 (we focus on the x dimension for simplicity).

In Fig. 7 we see the stiffness profiles before any via-points are added at $t > 1.0$. The left plot shows that after the skill ends at $t = 1.0$, the stiffness computed by inverting the covariance (3) decreases slowly to zero, an effect that is
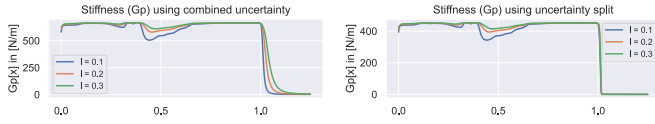
Fig. 7: Stiffness values $\boldsymbol{G}_P$ for the x dimension, before via-points are added. **Left:** Stiffness computed using the original KMP covariance (3), through $\boldsymbol{G}_P = (\text{cov}\,[\boldsymbol{\xi}(\boldsymbol{s}^*)])^{-1}$. **Right:** Stiffness computed using our approach that separates epistemic and aleatoric uncertainties (5). The different line colors correspond to different kernel lengths l, which we compared.
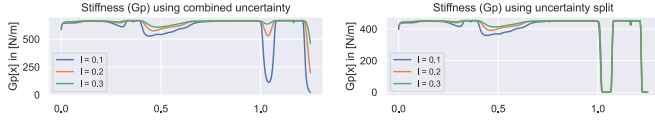

Fig. 8: Stiffness values $\boldsymbol{G}_P$ for the x dimension, after via-points are added. Left and right plots show the curves obtained with uncertainties computed from (3) and (4), respectively, similarly to Fig. 7, using different kernel length values l. Our approach (using uncertainty split) successfully makes the robot compliant in regions without data.

more noticeable as the kernel length l increases. This behavior explains the increased force profiles that we observed in Fig. 5, bottom left, since the robot collides with the box for a few instants as the stiffness approaches zero, making it unsafe to interact with. The right plot shows how our proposed approach (5) makes the stiffness approach zero faster and with lower sensitivity to the value of l. The parameter c_2 in the activation function allows regulation of the epistemic uncertainty threshold at which $\boldsymbol{G}_P$ transitions from being governed by aleatoric uncertainty to epistemic uncertainty. For this reason, even minor increases in epistemic uncertainty are sufficient to trigger a decrease in stiffness making the robot compliant and resulting in the lower interaction forces seen in Fig. 5, bottom right. Due to the continuity of the sigmoid function, the transition happens without discontinuities in the generated control actions. Figure 8 shows that after via-points are added, our approach successfully makes the robot compliant in regions without data, regardless of the kernel length scale, demonstrating improved compliance and efficiency.

REFERENCES

[1] M. Iskandar, C. Ott, O. Eiberger, M. Keppler, A. Albu-Schaffer, and A. Dietrich, "Joint-level control of the DLR lightweight robot SARA," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 8903–8910.

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

[3] 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.

[4] 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. Special issue on robot learning by observation, demonstration and imitation*, vol. 37, no. 2, 2007.

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

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

[7] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.

[8] Y. Huang, L. Rozo, J. Silvério, and D. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research (IJRR)*, vol. 38, pp. 833–852, 05 2019.

[9] Y. Huang, J. Silvério, L. Rozo, and D. G. Caldwell, "Generalized task-parameterized skill learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5667–5674.

[10] J. Zhu, M. Gienger, and J. Kober, "Learning task-parameterized skills from few demonstrations," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 4063–4070, 2022.

[11] J. Sun, J. Kober, M. Gienger, and J. Zhu, "Learning from few demonstrations with frame-weighted motion generation," in *18th International Symposium on Experimental Robotics (ISER)*, 2023, pp. 32–41.

[12] A. Sena, B. Michael, and M. Howard, "Improving task-parameterised movement learning generalisation with frame-weighted trajectory generation," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2019, pp. 4281–4287.

[13] J. Hoyos, F. Prieto, G. Alenyà, and C. Torras, "Incremental learning of skills in a task-parameterized gaussian mixture model," *Journal of Intelligent & Robotic Systems*, vol. 82, pp. 81–99, 2016.

[14] X. Yao, Y. Chen, and B. Tripp, "Improved generalization of probabilistic movement primitives for manipulation trajectories," *IEEE Robotics and Automation Letters (RA-L)*, vol. 9, no. 1, pp. 287–294, 2024.

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

[16] G. Franzese, C. Celemin, and J. Kober, "Learning interactively to resolve ambiguity in reference frame selection," in *CoRL*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 1298–1311.

[17] M. Saveriano and D. Lee, "Incremental skill learning of stable dynamical systems," in *Proc. IEEE/RSJ IROS*, 2018, pp. 6574–6581.

[18] M. Khoramshahi and A. Billard, "A dynamical system approach for detection and reaction to human guidance in physical human–robot interaction," *Autonomous Robots*, vol. 44, no. 8, pp. 1411–1429, 2020.

[19] S. Jauhri, C. Celemin, and J. Kober, "Interactive imitation learning in state-space," in *CoRL*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds. PMLR, 16–18 Nov 2021.

[20] G. Franzese, A. Meszaros, L. Peternel, and J. Kober, "Ilosa: Interactive learning of stiffness and attractors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[21] S. Jadav, J. Heidersberger, C. Ott, and D. Lee, "Shared autonomy via variable impedance control and virtual potential fields for encoding human demonstrations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 151–15 157.

[22] D. P. Losey and M. K. O'Malley, "Trajectory deformations from physical human–robot interaction," *IEEE Transactions on Robotics (T-RO)*, vol. 34, no. 1, pp. 126–138, 2018.

[23] J. Silvério, Y. Huang, L. Rozo, F. Abu-Dakka, and D. G. Caldwell, "Uncertainty-aware imitation learning using kernelized movement primitives," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 90–97.

[24] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1322–1328 vol.2.

[25] J. R. Medina and S. Hirche, "Uncertainty-dependent optimal control for robot control considering high-order cost statistics," in *IROS*, 2015.

[26] R. Calandra, A. Seyfarth, J. Peters, and M. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence (AMAI)*, vol. 76, 06 2015.

[27] S. Calinon, D. Bruno, and D. G. Caldwell, "A task-parameterized probabilistic model with minimal intervention control," in *ICRA*, 2014.

[28] J. Ramon Medina, D. Lee, and S. Hirche, "Risk-sensitive optimal feedback control for haptic assistance," in *ICRA*. IEEE, 2012.

[29] Y. Fanger, J. Umlauft, and S. Hirche, "Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 3913–3919.

[30] J. Y. Zhang and A. D. Dragan, "Learning from extrapolated corrections," in *Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 7034–7040.

[31] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Learning from interventions: Human-robot interaction as both explicit and implicit feedback," in *16th Robotics: Science and Systems (RSS)*. MIT Press Journals, 2020.

[32] A. Meszaros, G. Franzese, and J. Kober, "Learning to pick at non-zero-velocity from interactive demonstrations," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6052–6059, Jul. 2022.

[33] E. Todorov, "Optimality principles in sensorimotor control," *Nature Neuroscience*, vol. 7, no. 9, pp. 907–915, 2004.

[34] T. Kulak, H. Girgin, J.-M. Odobez, and S. Calinon, "Active learning of bayesian probabilistic movement primitives," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 2163–2170, 2021.

[35] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[36] C. Ott, A. Albu-Schaffer, A. Kugi, and G. Hirzinger, "On the passivity-based impedance control of flexible joint robots," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 416–429, Apr. 2008.

TABLE IV: Description of key notations

$\mathcal{I} \in \mathbb{N}$	$\triangleq$	Input dimension
$\mathcal{O} \in \mathbb{N}$	$\triangleq$	Output dimension
$M \in \mathbb{N}$	$\triangleq$	Number of demonstrations
$H \in \mathbb{N}$	$\triangleq$	Number of data points per demonstration
$N \in \mathbb{N}$	$\triangleq$	Number of gaussians per KMP
$\boldsymbol{s} \in \mathbb{R}^{\mathcal{I}}$	$\triangleq$	Input variable
$\boldsymbol{\xi} \in \mathbb{R}^{\mathcal{O}}$	$\triangleq$	Output variable
$\{\{\boldsymbol{s}_{h,m}, \boldsymbol{\xi}_{h,m}\}_{h=1}^{H}\}_{m=1}^{M}$	$\triangleq$	Set of demonstrations
$\boldsymbol{x}$	$\triangleq$	End-effector position in Cartesian space
$P \in \mathbb{N}$	$\triangleq$	Number of frames in a TP-KMP
$p = 1, \ldots, P$	$\triangleq$	Frame index
p^*	$\triangleq$	closest frame to a given via-point
$\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}$	$\triangleq$	Task parameters of frame p
$\boldsymbol{s}^{(p)}, \boldsymbol{\xi}^{(p)}$	$\triangleq$	Demonstrations represented locally in frame p
$\boldsymbol{\mu}^{(p)}, \boldsymbol{\Sigma}^{(p)}$	$\triangleq$	Local mean and covariance of frame p
$\hat{\boldsymbol{\mu}}^{(p)}, \hat{\boldsymbol{\Sigma}}^{(p)}$	$\triangleq$	Mean and covariance of frame p in global frame
$\bar{v} = \{\bar{s}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$	$\triangleq$	Via-point in global frame
$\bar{v}^{(p)} = \{\bar{s}^{(p)}, \bar{\boldsymbol{\mu}}^{(p)}, \bar{\boldsymbol{\Sigma}}^{(p)}\}$	$\triangleq$	Via-point in local frame p
$\boldsymbol{D}^{(p)} = \{\boldsymbol{s}_n^{(p)}, \boldsymbol{\mu}_n^{(p)}, \boldsymbol{\Sigma}_n^{(p)}\}_{n=1}^{N}$	$\triangleq$	Local reference trajectory distribution
$\Theta^{(p)} = \{\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}, \boldsymbol{D}^{(p)}\}$	$\triangleq$	Local KMP
$\Theta = \{\Theta^{(p)}\}_{p=1}^{P}$	$\triangleq$	TP-KMP
λ_1, λ_2	$\triangleq$	Regularization terms for KMP mean and covariance
α	$\triangleq$	Scaling factor for KMP covariance
l	$\triangleq$	Length scale of the kernel
$k(.,.)$	$\triangleq$	Kernel function
$\boldsymbol{K}, \boldsymbol{k}^*, \boldsymbol{k}^{**}$	$\triangleq$	Kernel matrices for different combinations of inputs
$\boldsymbol{K}_{(p)}, \boldsymbol{k}_{(p)}^*, \boldsymbol{k}_{(p)}^{**}$	$\triangleq$	p th local KMP kernel matrices
$\boldsymbol{\Sigma}_{\text{ep}}^*, \boldsymbol{\Sigma}_{\text{al}}^*$	$\triangleq$	Epistemic and aleatoric terms of KMP covariance
σ_{ep}^2	$\triangleq$	Epistemic variance (diagonal element of $\boldsymbol{\Sigma}_{\text{ep}}^*$)
$\boldsymbol{F}$	$\triangleq$	External force measured at the end-effector
γ_F	$\triangleq$	Threshold for triggering force-based via-points
γ_ξ	$\triangleq$	Threshold for triggering trajectory-based via-points
γ_Σ	$\triangleq$	Prior covariance of new via-points
γ_D	$\triangleq$	Prior covariance of *placeholder* KMP
w_1, w_2	$\triangleq$	Weights of epistemic and aleatoric uncertainty
c_1, c_2	$\triangleq$	Sigmoid function parameters
$\delta_{\text{ep}}, \delta_{\text{al}}$	$\triangleq$	Epistemic and aleatoric scaling factors
$\boldsymbol{G}_P, \boldsymbol{G}_D$	$\triangleq$	Stiffness and damping gains

TABLE V: Glossary of important acronyms

LfD	$\triangleq$	Learning from Demonstration
DoF	$\triangleq$	Degrees of Freedom
TP	$\triangleq$	Task Parameterization
(TP)-GMM	$\triangleq$	(Task parameterized)- Gaussian Mixture Model [2]
(TP)-GMR	$\triangleq$	(Task parameterized)- Gaussian Mixture Regression [2]
(TP)-KMP	$\triangleq$	(Task parameterized)- Kernelized Movement Primitive [8]
DMP	$\triangleq$	Dynamic Movement Primitives [3]
ProMP	$\triangleq$	Probabilistic Movement Primitives [15]
LfEC	$\triangleq$	Learning from Extrapolated Corrections [30]
LfI	$\triangleq$	Learning from Interventions [31]

A. Key notations

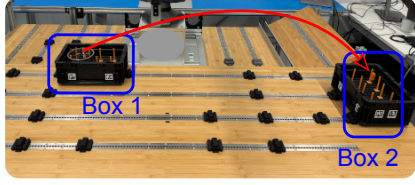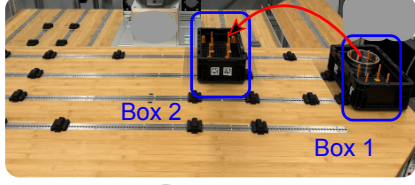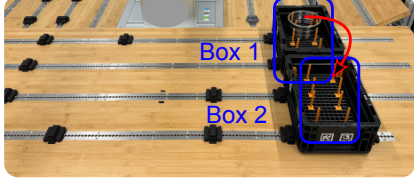Table IV summarizes the key notations used in our framework.

B. Key acronyms

Table V shows a glossary of important acronyms in our approach.

C. Evaluation details

Table VI and Table VII show more details of the quantitative evaluation of 15 different task-parameterized scenarios, mentioned in Sec. V-B of the paper.

TABLE VI: (PART I) Quantitative evaluation of 15 different task parameterized scenarios. ✓successful, ✓successful but contact force was above 10 N, ✗failure, ✗* successful but only after correcting the whole trajectory, which we count as failure.

#	Configuration	TP-GMM	*vanilla* TP-KMP	Ours
1		✓	✓	✓
2		✓	✓	✓
3		✓	✓	✓
4		✗	✗	✓
5		✗	✗	✗*
6		✗	✗	✓
7		✓	✓	✓
8		✓	✓	✓

TABLE VII: (PART II) Quantitative evaluation of 15 different task parameterized scenarios. ✓ successful, ✓ successful but contact force was above 10 N, ✗ failure, ✗* successful but only after correcting the whole trajectory, which we count as failure.

#	Configuration	TP-GMM	*vanilla* TP-KMP	Ours
9		✗	✗	✓
10		✗	✗	✓
11		✓	✓	✓
12		✓	✓	✓
13		✓	✓	✓
14		✓	✓	✓
15		✗	✗	✓
Success rate:		60 %	60 %	93.33 %