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]
[1] German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC),
Münchener Str. 20, 82234 Weßling, Germany
[2] Technical University of Munich (TUM), School of Computation, Information and Technology (CIT)
Arcisstr. 21, 80333 Munich, Germany
markus.knauer@dlr.de

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 two prominent approaches have emerged. While one leverages via-points to adapt skills locally by modulating demonstrated trajectories, the other relies on so-called *task-parameterized* 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, robot [1].

Keywords: task-parameterized imitation learning, interactive learning

1 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 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,

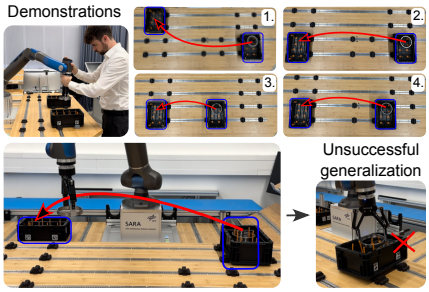


Figure 1: By demonstrating on the same table, users inadvertently introduce low vertical axis variance in multiple demonstrations of a ring-loading task, causing TP-GMM [2] to fail to generalize to different table heights.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

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.

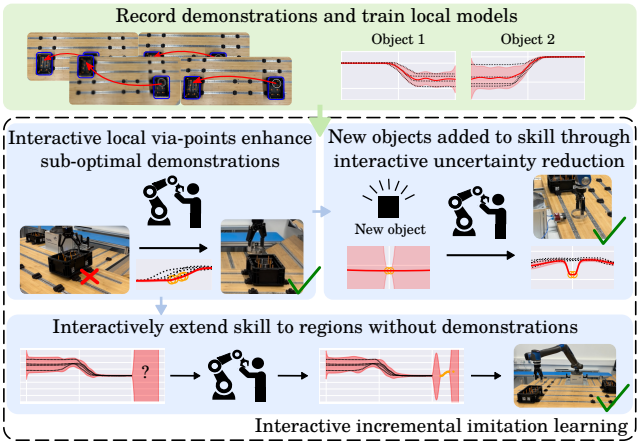


Figure 2: Approach overview. Users provide demonstrations fitted locally to different objects using KMPs (top). The resulting model, TP-KMP, is interactively adjusted to improve generalization and augment the demonstrated skill. An interactive strategy to add via-points locally, with respect to different objects, is the cornerstone of our approach, enabling correction of skills (center-left), addition of new objects (center-right), and skill augmentation (bottom).

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 [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. 3 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-parameterized skills with local trajectory modulation (Sec. 4) 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. 5) 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 Sections 6–7). Appendices A.1–A.2 summarize notations and acronyms.

2 Related work

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 some scenarios, they do not permit adding new task parameters to existing models and, most importantly, they do not provide interactive capabilities to incrementally refine skills.

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, 18, 19, 20, 21], where skills are learned as time-independent autonomous systems and modulated directly in task space by introducing new state-action pairs. In 2019, 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.

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 assigns high importance to the corresponding objects when applied to local models, favoring via-points in new situations. 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 with large variance and improving skills incrementally. This is based on epistemic uncertainty, as shown in [23], where KMPs compute both aleatoric and epistemic uncertainties—randomness in data and model knowledge gaps, respectively [7]. Uncertainty quantifications, essential in robotics [24, 25, 26], allow us to define robot behaviors through uncertainty-aware stiffness regulation strategies [27, 28], similar to Bayesian mixture models [29, 30].

3 Preliminaries

Let us denote a set of demonstrations by $\{\{\boldsymbol{s}_{h,m}, \boldsymbol{\xi}_{h,m}\}_{h=1}^{H}\}_{m=1}^{M}$, where $\boldsymbol{s} \in \mathbb{R}^{\mathcal{I}}$ and $\boldsymbol{\xi} \in \mathbb{R}^{\mathcal{O}}$ 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 $\boldsymbol{s}$ and $\boldsymbol{\xi}$ from demonstrations. Depending on the task, $\boldsymbol{s}$ is often time or robot state while $\boldsymbol{\xi}$ represents desired poses or velocities. In Sec. 3 and Sec. 4 we keep these generic except when concrete examples help explain new concepts.

Task-parameterized movement models. In TP models [2] a *frame* $p = 1, \dots, P$ is described by so-called *task parameters* $\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}$, 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 $\boldsymbol{\xi}^{(p)} = \boldsymbol{A}^{(p)^{-1}}\left(\boldsymbol{\xi} - \boldsymbol{b}^{(p)}\right)$. Local datasets are modeled probabilistically yielding a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^{(p)}, \boldsymbol{\Sigma}^{(p)})$ for every input $\boldsymbol{s}$. During skill execution, local distributions are mapped to the common frame, as both task parameters and inputs change with time t, through $\hat{\boldsymbol{\mu}}_t^{(p)} = \boldsymbol{A}_t^{(p)}\boldsymbol{\mu}_t^{(p)} + \boldsymbol{b}_t^{(p)}$, $\hat{\boldsymbol{\Sigma}}_t^{(p)} = \boldsymbol{A}_t^{(p)}\boldsymbol{\Sigma}_t^{(p)}\boldsymbol{A}_t^{(p)^\top}$. Values of $\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}$ may differ from those seen during the demonstrations, and, through $\hat{\boldsymbol{\mu}}_t^{(p)}$, $\hat{\boldsymbol{\Sigma}}_t^{(p)}$, each frame provides a model of $\boldsymbol{\xi}$ from its perspective. The distributions from different coordinate systems are fused by a product of Gaussians, resulting in a new distribution $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, in the common frame, with parameters

$$\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t \sum_{p=1}^{P} \hat{\boldsymbol{\Sigma}}_t^{(p)^{-1}} \hat{\boldsymbol{\mu}}_t^{(p)}, \;\; \boldsymbol{\Sigma}_t = \left(\sum_{p=1}^{P} \hat{\boldsymbol{\Sigma}}_t^{(p)^{-1}}\right)^{-1}. \tag{1}$$

[1]Task parameters can represent a number of affine transformations commonly found in robotics [2]. Here, as in other works [9, 10, 11, 14], we use them to represent the coordinate systems of objects of interest.

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.

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

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

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

where $\boldsymbol{K} = [\hat{\boldsymbol{k}}(\boldsymbol{s}_1)^\top, \dots, \hat{\boldsymbol{k}}(\boldsymbol{s}_N)^\top]$, $\boldsymbol{k}^* = \hat{\boldsymbol{k}}(\boldsymbol{s}^*)$, with $\hat{\boldsymbol{k}}(\boldsymbol{s}_i) = [\boldsymbol{k}(\boldsymbol{s}_i, \boldsymbol{s}_1), \dots, \boldsymbol{k}(\boldsymbol{s}_i, \boldsymbol{s}_N)]$, $\boldsymbol{k}^{**} = \boldsymbol{k}(\boldsymbol{s}^*, \boldsymbol{s}^*)$, $\boldsymbol{k}(\boldsymbol{s}_i, \boldsymbol{s}_j) = k(\boldsymbol{s}_i, \boldsymbol{s}_j)\boldsymbol{I}$ and $k(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is a kernel function. Moreover, $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1^\top \dots \boldsymbol{\mu}_N^\top\right]^\top$, $\boldsymbol{\Sigma} = \operatorname{blockdiag}\left(\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_N\right)$ and λ_1, λ_2, α are hyperparameters. The kernel matrices are denoted as $\boldsymbol{K}, \boldsymbol{k}^*$ and $\boldsymbol{k}^{**}$. From (2)–(3) it follows that if, for a certain $\boldsymbol{\mu}_n$, the covariance $\boldsymbol{\Sigma}_n$ is small, the expectation at $\boldsymbol{s}_n$ will be close to $\boldsymbol{\mu}_n$. This provides a principled way for trajectory modulation. Indeed, if, for a new input $\bar{\boldsymbol{s}}$, one wants to ensure that the expectation passes through a desired $\bar{\boldsymbol{\mu}}$, it suffices to manually add the pair $\{\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ to the reference distribution provided that $\bar{\boldsymbol{\Sigma}}$ is small enough. This both makes (2) closely match $\bar{\boldsymbol{\mu}}$ and lowers the covariance (3) to match $\bar{\boldsymbol{\Sigma}}$.

4 Interactive local trajectory modulation with TP-KMP

Similarly to [8], we define a local KMP as a model $\Theta^{(p)} = \{\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}, \boldsymbol{D}^{(p)}\}$ with associated task parameters $\boldsymbol{b}^{(p)}$, $\boldsymbol{A}^{(p)}$ and $\boldsymbol{D}^{(p)} = \{\boldsymbol{s}_n^{(p)}, \boldsymbol{\mu}_n^{(p)}, \boldsymbol{\Sigma}_n^{(p)}\}_{n=1}^N$. $\boldsymbol{\mu}^{(p)}$, $\boldsymbol{\Sigma}^{(p)}$ are computed from output data projected locally $\boldsymbol{\xi}^{(p)}$, 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 $\mathcal{N}(\boldsymbol{\mu}^{(p)}, \boldsymbol{\Sigma}^{(p)})$, 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.

4.1 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*

Define external force threshold γ_F, distance threshold γ_ξ and via-point variance γ_Σ
Input: P local KMPs $\Theta^{(p)} = \{\boldsymbol{b}^{(p)}, \boldsymbol{A}^{(p)}, \boldsymbol{D}^{(p)}\}_{p=1}^P$ trained from $\{\{\boldsymbol{s}_{h,m}, \boldsymbol{\xi}_{h,m}\}_{h=1}^H\}_{m=1}^M$
for time-step t in all time-steps T **do**
  if interaction is triggered **then** ▷ Button press, force or distance trigger (see Section 4.1)
    $\bar{v}_t = \{\bar{\boldsymbol{s}} = t, \bar{\boldsymbol{\mu}} = \boldsymbol{\xi}_t, \bar{\boldsymbol{\Sigma}} = \gamma_\Sigma \boldsymbol{I}\}$ ▷ create global via-point
    $p^* = \operatorname{argmin}_p ||\bar{\boldsymbol{\mu}} - \boldsymbol{b}^{(p)}||$ ▷ Find p^* (see Section 4.1)
    $\bar{v}^{(p^*)} = \{\bar{\boldsymbol{s}}^{(p^*)}, \bar{\boldsymbol{\mu}}^{(p^*)}, \bar{\boldsymbol{\Sigma}}^{(p^*)}\}$ ▷ Map via-point $\bar{v}_t$ locally to p^* (see Section 3)
    $\boldsymbol{D}^{(p^*)} \leftarrow \boldsymbol{D}^{(p^*)} \cup \bar{v}^{(p^*)}$ ▷ Update local KMP p^* with via-point
Recompute $\boldsymbol{K}^{(p)}$, $\boldsymbol{k}_{(p)}^*$ and $\boldsymbol{k}_{(p)}^{**}$ for all p where via-points were added (see Sec. 3)

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:** as some collaborative robots have button interfaces at the end-effector, via-points can be triggered through button pressing.

In all cases, a via-point is defined globally, in a common frame, as $\bar{v}_t = \{\bar{\boldsymbol{s}} = t, \bar{\boldsymbol{\mu}} = \boldsymbol{\xi}_t, \bar{\boldsymbol{\Sigma}} = \gamma_\Sigma \boldsymbol{I}\}$, 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. Button presses are a flexible approach, however not all collaborative robots have such functionality. In Section 5, 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, $p^* = \text{argmin}_p ||\bar{\boldsymbol{\mu}}_t - \boldsymbol{b}^{(p)}||$. 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 Section 3 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.

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 $\boldsymbol{D}^{(P+1)}$: $\boldsymbol{D}^{(P+1)} = \{\boldsymbol{s}_N^{(P+1)} = \boldsymbol{s}_n^{(P)}, \boldsymbol{\mu}_n^{(P+1)} = \boldsymbol{0}, \boldsymbol{\Sigma}_n^{(P+1)} = \gamma_D \boldsymbol{I}\}_{n=1}^N$
- 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 Algorithm 1**

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.

4.2 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, 31]. In the case of epistemic uncertainty, such schemes contribute to improved safety [23]. 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 [30], the covariance prediction of KMP can be decomposed into two terms, corresponding to aleatoric and epistemic components:

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

See A.3 for the derivation. The term $\boldsymbol{\Sigma}^*_{\text{ep}}$ is the same as the variance prediction in Gaussian process regression [32], 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

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

where $w_1 \in [0, 1]$ is a sigmoid function that depends on the epistemic uncertainty $w_1(\sigma^2_{\text{ep}}) = 1/(1 + e^{-c_1(\sigma^2_{\text{ep}} - c_2)})$, with $\sigma^2_{\text{ep}} = \operatorname{tr}(\boldsymbol{\Sigma}^*_{\text{ep}})/\mathcal{O}$, 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 δ_{ep}, δ_{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, contributing to safety. 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, 31], in regions where data was provided.

5 Evaluation

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. We provide $M = 4$ demonstrations with different box positions, see Fig. 1-top and Appendix A.4.2. We use a time-driven representation with $\boldsymbol{s}_{h,m} = t_{h,m}/T_m$, 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 [32]). Other KMP hyperparameters were $\lambda_1 = 0.1$, $\lambda_2 = 1$, $\alpha = 1$, chosen empirically. 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). Also see Appendix A.4.3.

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 Appendix A.4.5).

[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. $\boldsymbol{G}_D$ is chosen empirically.

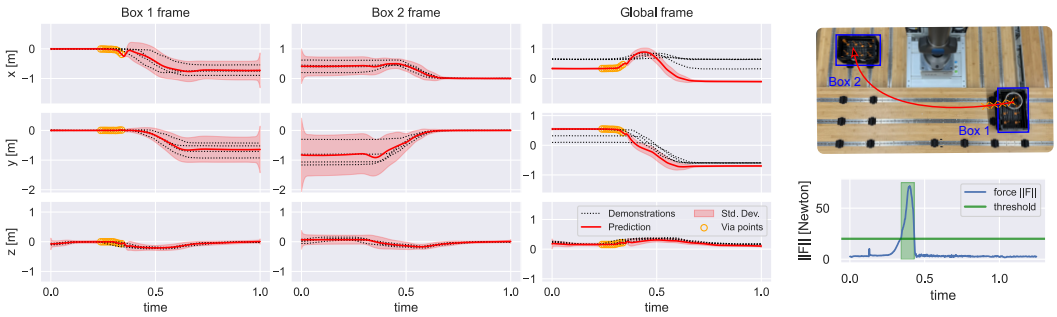


Figure 3: Improving generalization with interactively-defined local via-points. Since the robot would fail, a correction is shown close to box A, indicated by via-points in orange.

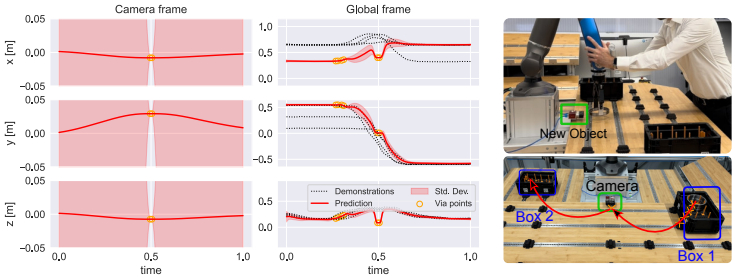


Figure 4: Interactively adding a new object to existing skill. We add a camera to the scene, associate it with a new frame and add via-points to pass in front of the camera, this time by pressing a button.

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). See A.4.6 for generalization results to a new camera location.

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 $c_1 = 5 \times 10^3$, $c_2 = 1.5 \times 10^{-3}$, $\delta_{\text{ep}} = 1 \times 10^3$, $\delta_{\text{al}} = 1$, chosen empirically. We consider a baseline where the gains are computed from the original KMP covariance (3) as $\boldsymbol{G}_P = (\text{cov}\,[\boldsymbol{\xi}(\boldsymbol{s}^*)])^{-1}$. To address numerical issues and ensure a maximum stiffness value we regularize $\text{cov}\,[\boldsymbol{\xi}(\boldsymbol{s}^*)]$, $\delta_{\text{al}}\boldsymbol{\Sigma}^*_{\text{al}}$, $\delta_{\text{ep}}\boldsymbol{\Sigma}^*_{\text{ep}}$ 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 [32]). With the improved safety 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

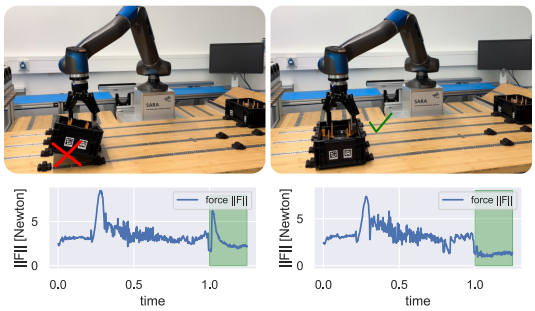


Figure 5: **Left:** Computing stiffness gains from (3) leads to increased interaction forces at non-demonstrated inputs. **Right:** Using our approach (4)–(5) the robot reacts quickly to the increase in the epistemic uncertainty, lowering its stiffness.

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. Appendices A.4.7, A.4.8 show the global model resulting from Experiments 1–3 and the detailed stiffness profiles from Experiment 3 respectively.

6 Discussion

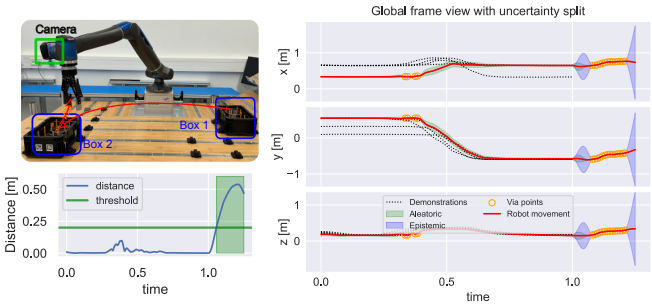


Figure 6: Adding via-points at $t > 1.0$.

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 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 as well (Fig. 6-right). Also see Fig. 13.

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 defining via-points for inputs where via-points have already been added. This makes our approach better tailored to building on libraries of pre-trained models (without via-points) that can be interactively refined and adapted to new tasks, rather than building on the same model over long periods of time. To provide users with more options for skill re-use, in future work we will investigate mechanisms to interactively remove via-points.

7 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 learning from demonstration (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.

Acknowledgments

This work was partially funded by the DLR project "Factory of the Future Extended" and the European Union's Horizon Research and Innovation Programme under Grant 101136067 (INVERSE).

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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2020. doi:10.1109/iros45743.2020.9340700. 1

[2] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, 2016. 1, 2, 3, 6, 13, 14, 16

[3] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013. 1, 13, 14

[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*, 37(2):286–298, 2007. 1, 3

[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*, 3:297–330, 2020. 1, 13

[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*, 10(1-2):1–197, 2022. 2, 3

[7] E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. ISSN 1573-0565. doi:10.1007/s10994-021-05946-3. 2, 3

[8] Y. Huang, L. Rozo, J. Silvério, and D. Caldwell. Kernelized movement primitives. *The International Journal of Robotics Research (IJRR*, 38:833–852, 05 2019. doi:10.1177/0278364919846363. 2, 3, 4, 6, 13, 14, 16

[9] Y. Huang, J. Silvério, L. Rozo, and D. G. Caldwell. Generalized task-parameterized skill learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5667–5674, 2018. 2, 3

[10] J. Zhu, M. Gienger, and J. Kober. Learning task-parameterized skills from few demonstrations. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):4063–4070, 2022. 2, 3

[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. 2, 3

[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)*, pages 4281–4287, 2019. 2

[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*, 82:81–99, 2016. 2

[14] X. Yao, Y. Chen, and B. Tripp. Improved generalization of probabilistic movement primitives for manipulation trajectories. *IEEE Robotics and Automation Letters (RA-L)*, 9(1):287–294, 2024. 2, 3, 14, 15, 16

[15] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. *Advances in neural information processing systems (NeurIPS)*, 26, 2013. 2, 3, 13, 14

[16] G. Franzese, C. Celemin, and J. Kober. Learning interactively to resolve ambiguity in reference frame selection. In J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning (CoRL)*, volume 155 of *Proceedings of Machine Learning Research*, pages 1298–1311. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/franzese21a.html. 3

[17] M. Saveriano and D. Lee. Incremental skill learning of stable dynamical systems. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 6574–6581, 2018. 3

[18] M. Khoramshahi and A. Billard. A dynamical system approach for detection and reaction to human guidance in physical human–robot interaction. *Autonomous Robots*, 44(8):1411–1429, 2020. 3

[19] S. Jauhri, C. Celemin, and J. Kober. Interactive imitation learning in state-space. In J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning (CoRL)*, volume 155 of *Proceedings of Machine Learning Research*, pages 682–692. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/jauhri21a.html. 3

[20] G. Franzese, A. Mészáros, L. Peternel, and J. Kober. Ilosa: Interactive learning of stiffness and attractors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7778–7785, 2021. 3

[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. 3

[22] D. P. Losey and M. K. O'Malley. Trajectory deformations from physical human–robot interaction. *IEEE Transactions on Robotics (T-RO)*, 34(1):126–138, 2018. 3

[23] J. Silvério, Y. Huang, L. Rozo, F. Abu-Dakka, and D. G. Caldwell. Uncertainty-aware imitation learning using kernelized movement primitives. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. 3, 5, 6

[24] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 343–349, 01 1999. 3

[25] J. R. Medina and S. Hirche. Uncertainty-dependent optimal control for robot control considering high-order cost statistics. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015. doi:10.1109/iros.2015.7353940. 3

[26] R. Calandra, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence (AMAI)*, 76, 06 2015. doi:10.1007/s10472-015-9463-9. 3

[27] S. Calinon, D. Bruno, and D. G. Caldwell. A task-parameterized probabilistic model with minimal intervention control. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014. doi:10.1109/icra.2014.6907339. 3

[28] J. Ramon Medina, D. Lee, and S. Hirche. Risk-sensitive optimal feedback control for haptic assistance. In *2012 IEEE International Conference on Robotics and Automation(ICRA)*. IEEE, 2012. doi:10.1109/icra.2012.6225085. 3, 5, 6

[29] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Secaucus, NJ, USA, 2006. 3

[30] T. Kulak, H. Girgin, J.-M. Odobez, and S. Calinon. Active learning of bayesian probabilistic movement primitives. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):2163–2170, 2021. 3, 6, 13

[31] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9):907–915, 2004. 5, 6

[32] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. 6, 7, 13

[33] J. Y. Zhang and A. D. Dragan. Learning from extrapolated corrections. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019. doi:10.1109/icra.2019.8793554. 13, 14

[34] 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 2020*. MIT Press Journals, 2020. 13, 14

[35] N. Jaquier, D. Ginsbourger, and S. Calinon. Learning from demonstration with model-based gaussian process. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *Proceedings of the Conference on Robot Learning (CoRL)*, volume 100 of *Proceedings of Machine Learning Research*, pages 247–257. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/jaquier20b.html. 14

A Additional material

A.1 Key notations

Table 1 summarizes the key notations used in our framework.

Table 1: 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^2_{ep}	$\triangleq$	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
δ_{ep}, δ_{al}	$\triangleq$	Epistemic and aleatoric scaling factors
$\boldsymbol{G}_P$, $\boldsymbol{G}_D$	$\triangleq$	Stiffness and damping gains

A.2 Key acronyms

Table 2 shows a glossary of important acronyms in our approach.

Table 2: 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 [33]
LfI	$\triangleq$	Learning from Interventions [34]

A.3 Decomposition of KMP covariance into a sum of epistemic and aleatoric terms

We here show that, similarly to [30], the covariance prediction of KMP can be decomposed into two distinct terms, corresponding to aleatoric and epistemic components. Using the Woodbury identity $(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{B}\boldsymbol{V})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{B}^{-1}+\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}$ with $\boldsymbol{V} = \boldsymbol{U} = \boldsymbol{I}$, we can rewrite (3) as (omitting α for the sake of the derivation):

$$\begin{aligned}
\operatorname{cov}\left[\boldsymbol{\xi}(\boldsymbol{s}^*)\right] &= \boldsymbol{k}^{**} - \boldsymbol{k}^*\left[\boldsymbol{K}^{-1} - \boldsymbol{K}^{-1}\left(\boldsymbol{K}^{-1}+(\lambda\boldsymbol{\Sigma})^{-1}\right)^{-1}\boldsymbol{K}^{-1}\right]\boldsymbol{k}^{*\top} && (6)\\
&= \boldsymbol{k}^{**} - \boldsymbol{k}^*\boldsymbol{K}^{-1}\boldsymbol{k}^{*\top} + \boldsymbol{k}^*\boldsymbol{K}^{-1}\left(\boldsymbol{K}^{-1}+(\lambda\boldsymbol{\Sigma})^{-1}\right)^{-1}\boldsymbol{K}^{-1}\boldsymbol{k}^{*\top} && (7)\\
&= \underbrace{\boldsymbol{k}^{**} - \boldsymbol{k}^*\boldsymbol{K}^{-1}\boldsymbol{k}^{*\top}}_{\boldsymbol{\Sigma}^*_{\mathrm{ep}}} + \underbrace{\boldsymbol{k}^*\left(\boldsymbol{K}+\boldsymbol{K}(\lambda\boldsymbol{\Sigma})^{-1}\boldsymbol{K}\right)^{-1}\boldsymbol{k}^{*\top}}_{\boldsymbol{\Sigma}^*_{\mathrm{al}}}, && (8)
\end{aligned}$$

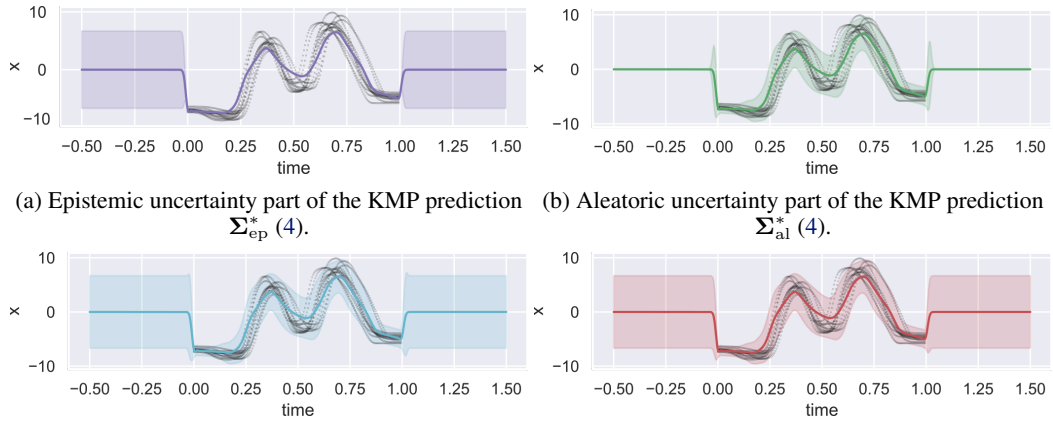
where, in the last step, we used $\boldsymbol{A}^{-1}\left(\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1}\right)^{-1}\boldsymbol{A}^{-1} = \left(\boldsymbol{A}+\boldsymbol{A}\boldsymbol{B}^{-1}\boldsymbol{A}\right)^{-1}$. We further drop the explicit dependence on $\boldsymbol{s}^*$, for simplicity, and use the superscript '$*$'. Note the clear separation between the two terms in (4). The term $\boldsymbol{\Sigma}^*_{\mathrm{ep}}$ is the same as the variance prediction in Gaussian process regression [32], corresponding to the epistemic part of the KMP covariance. The remaining term $\boldsymbol{\Sigma}^*_{\mathrm{al}}$ gives the aleatoric uncertainty.

For illustrative purposes, we trained a KMP on a 2D dataset with input data on the interval $[0; 1]$ and computed the mean and covariance using (2), (3) and (4) for the $[-0.5; 1.5]$ interval, including both terms $\boldsymbol{\Sigma}^*_{\mathrm{ep}}$ and $\boldsymbol{\Sigma}^*_{\mathrm{al}}$. In Fig. 7a and Fig. 7b we plot the epistemic and aleatoric terms, respectively. Figure 7c shows that their sum gives the same result the original KMP formulation (3), which can be seen in Fig. 7d.

A.4 Experimental details and additional evaluations

A.4.1 Approach comparison

In Table 3 we compare our approach to other task-parameterized approaches on feature level. For completeness we also compare to non-task-parameterized approaches. However, as shown in [5] task-parameterized approaches are more suitable for adapting to new situations, including those involving multiple objects. To the best of our knowledge, we are the first to present a task-parameterized approach allowing interactive modulations, extending of the skill with new frames and the application of local via-points, making it possible to move corrections with their corresponding objects. We are also unique in providing aleatoric as well as the epistemic uncertainties, which allow usage where those values are needed individually (like shown in Sec. 4.2). Since our approach offers the most features to adapt and extend motion primitives incrementally, we believe it shows the most potential for different application scenarios.



(a) Epistemic uncertainty part of the KMP prediction $\boldsymbol{\Sigma}^*_{\text{ep}}$ (4).

(b) Aleatoric uncertainty part of the KMP prediction $\boldsymbol{\Sigma}^*_{\text{al}}$ (4).

(c) KMP covariance prediction as $\text{cov}[\boldsymbol{\xi}(\boldsymbol{s}^*)] = \boldsymbol{\Sigma}^*_{\text{ep}} + \boldsymbol{\Sigma}^*_{\text{al}}$ (4).

(d) Original KMP covariance prediction [8], computed from (3).

Figure 7: The proposed KMP covariance prediction (4) is computed explicitly as a sum of epistemic and aleatoric terms. Training data are shown as dotted black lines. Solid lines represent means and shaded areas are uncertainties.

Table 3: Comparison of related approaches. Ours is the only one having both uncertainties and offers the most tools to adapt motion primitives. *available jointly but not used for interaction

	aleatoric uncertainty	epistemic uncertainty	global via-point	local via-point	new frames	interactive modulation
Task-parameterized approaches						
TP-GMM/GMR [2]	✓	-	-	-	-	-
TP-ProMP [14]	✓	-	✓	-	-	-
vanilla TP-KMP [8]	✓*	✓*	✓	-	-	-
Our approach	✓	✓	✓	✓	✓	✓
Non-task-parameterized approaches						
GP/GPR [35]	✓*	✓*	✓	-	-	-
DMP [3]	-	-	✓	-	-	-
LfEC [33]	-	-	✓	-	-	✓
LfI [34]	-	-	✓	-	-	✓

In Table 4 we compare our approach to other LfD approaches (with a strong focus on task-parameterized formulations) in terms of performance. For this comparison, we followed 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. TP-KMP [8] performs similarly to TP-ProMP [14]. In contrast, 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. Please note that we did not in-

Table 4: Mean ± 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 taken from [14]. The others 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	**48.51 ± 12.21**
TP-KMP [8]	39.30 ± 29.06	6.93 ± 5.08	51.95 ± 26.44
Our approach (with via-points)	$\mathbf{3.88 \times 10^{-4} \pm 2.54 \times 10^{-4}}$	$\mathbf{1.16 \times 10^{-3} \pm 6.38 \times 10^{-4}}$	50.23 ± 25.58

clude via-points besides at start and end since, similarly to [14], we did not consider other via-points necessary for improving precision in this task.

A.4.2 Demonstration recordings

We recorded four demonstrations, where the robot picks an inner ring of a ball bearing out of a box and places it inside another box, see Fig. 8 for the setup. We use different box poses for each demonstration and record the data by kinesthetically teaching a torque-controlled 7-DoF manipulator running a gravity compensation controller. Figure 9 shows the collected demonstrations, together with their local distributions from the perspective of frames 1 and 2, as well as the global frame (for the task parameters of demonstration 1).

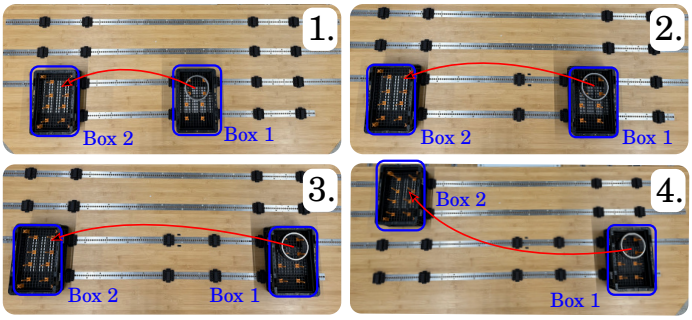


Figure 8: Overview of the four given demonstrations, where an inner ring of a ball bearing is transferred from one box to another. The poses of the boxes differ in each demonstration. Exemplary demonstration trajectories are indicated as a red line, moving from 'Box 1' to 'Box 2'.

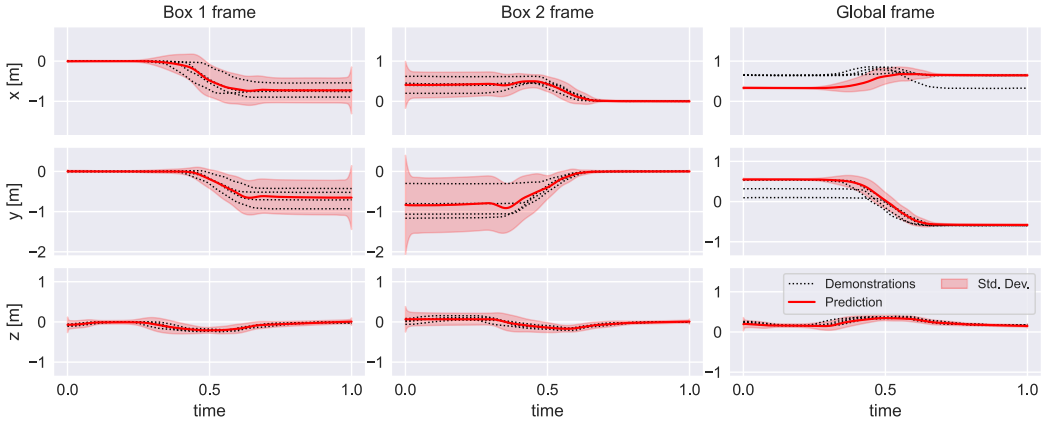


Figure 9: Recorded demonstrations in local and global frames. The demonstrations are shown as a black doted line. The red lines and shaded area represent the mean and one standard deviation. The distribution in the global frame is generated from the task parameters of demonstration 1.

A.4.3 Hyper-parameter overview

Table 5 shows the different hyperparameters used in this paper and a tested range of values to provide a guideline for choosing hyperparameters. We also explained what an increase or decrease of the individual parameter would lead to. In practice, the choice of parameters involves trade-offs between smoothness and precision in the robot's movement (owing largely to the kernel-based nature of our approach). This is evident in our findings (see Fig. 14–15), which demonstrate how different parameter settings can affect the robot's performance without necessarily impacting success or failure.

Table 5: Hyper-parameter overview

Hyper-parameter	Used in paper	Validated range/options working	Remark
Amount of gaussians N	12	7 – 40	$\uparrow \mathbb{N}$: for more complex and/or longer tasks
Regularization term KMP mean λ_1	0.1	0.1 – 1.0	$\uparrow \lambda_1$: for closer following the mean distribution of the KMP
Regularization term KMP variance λ_2	1.0	0.1 – 1.0	$\uparrow \lambda_2$: for 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 accuracte
Kernel function $k(.,.)$	matern2	matern2 / rbf	-
Sigmoid function parameter c_1	5×10^3	$1 - 1 \times 10^4$	$\uparrow c_1$: control gains switch more quickly between aleatoric and epistemic uncertainty
Sigmoid function parameter c_2	1.5×10^{-3}	$1 \times 10^{-4} - 1$	$\uparrow c_2$: epistemic uncertainty takes over for larger values of σ^2_{ep}
Epistemic scaling factor δ_{ep}	1×10^3	$1 - 1 \times 10^4$	$\uparrow \delta_{\text{ep}}$: increases the influence of epistemic uncertainty
Aleatoric scaling factor δ_{al}	1	$1 - 1 \times 10^4$	$\uparrow \delta_{\text{al}}$: increases the influence of aleatoric uncertainty

A.4.4 KMP hyperparameter sensitivity analysis

We examine the impact of KMP hyperparameter choices in Section 5 on the generated trajectories and associated uncertainties. For a more comprehensive analysis, readers are referred to the original source [8]. In the following, when varying parameters, the remaining ones are kept to the values used in 5.

Figure 10 shows the effect of l and λ_1 in the mean trajectory. Lower values of l tend to overfit the data, resulting in less smooth trajectories, an effect we can see in 10–left. A similar effect occurs when decreasing λ_1, which acts as a regularization term in KMP solution (see [8] for details).

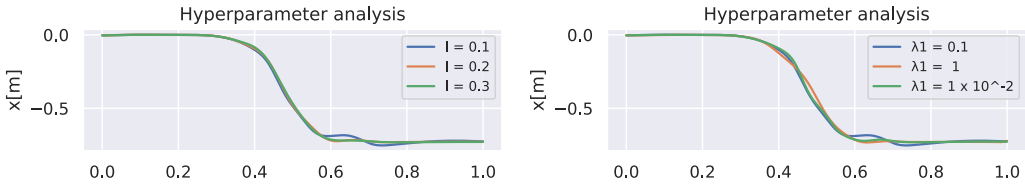


Figure 10: Cartesian position x from Box 1 frame (Fig.9, top-left). The different line colors correspond to different length scales l (**left**) and λ_1 (**right**), which we compared.

In Fig. 11–left we see the effect of varying λ_2. Similarly to λ_1, decreasing λ_2 tends to lead to a poorer fit of the observed covariances. Figure 11–right shows the effect of the factor α, which scales up/down the covariance to further improve the fit to the data. Note that, in Section 5, we used $\alpha = 1$, as done in [8].

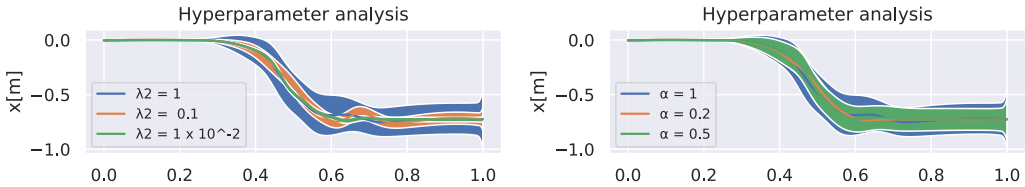


Figure 11: Cartesian position x from Box 1 frame. The different line colors correspond to different λ_2 (**left**) and re-scaling factor α (**right**), which we compared.

A.4.5 Generalization success rate for different conditions of the ring-loading task

Table 6 shows the individual successes and failures, as well as the success rates, for various box locations in the ring-loading task, comparing the performance of TP-GMM [2], TP-KMP [8] and our approach (TP-KMP with via-points from Experiment 1). Scenarios 5 and 15 were created by the authors, informed by their experience with scenarios where task-parameterized approaches may underperform. All the other 13 conditions were created in a user-informed manner by asking colleagues familiar with the workbench about configurations used in previous experiments and related projects. The aim of this section is to show how our contributions improve upon our baseline (TP-KMP) in various generalization tasks, therefore we did not compare to other approaches such as TP-ProMP [14] (a comparison on feature level can be found in Table 3).

We treated any collisions with the environment that exceeded a pre-defined force threshold as failures. Notably, our method achieves a 93% success rate, compared to 60% with TP-GMM/TP-KMP. While our approach generally excels, there is one notable exception (Configuration #5), where a failure occurs due to a scenario that is significantly different from the demonstrations. In all demonstrations, the movement proceeds from left to right, while Configuration #5 would require the movement to be entirely reversed. By defining additional via-points, it was ultimately possible to achieve success using our approach. Nevertheless, we argue that if the task significantly deviates from what was demonstrated, a new set of demonstrations should be provided.

A.4.6 Generalization of camera via-points to new camera location

Figure 12 shows the generalization of the via-points added to the camera frame in Experiment 2 to a new camera location. Notice how, for instance, the x coordinate of the via-points in the global frame increases to match the new camera location, which is farther away from the robot base than that in Fig. 4.

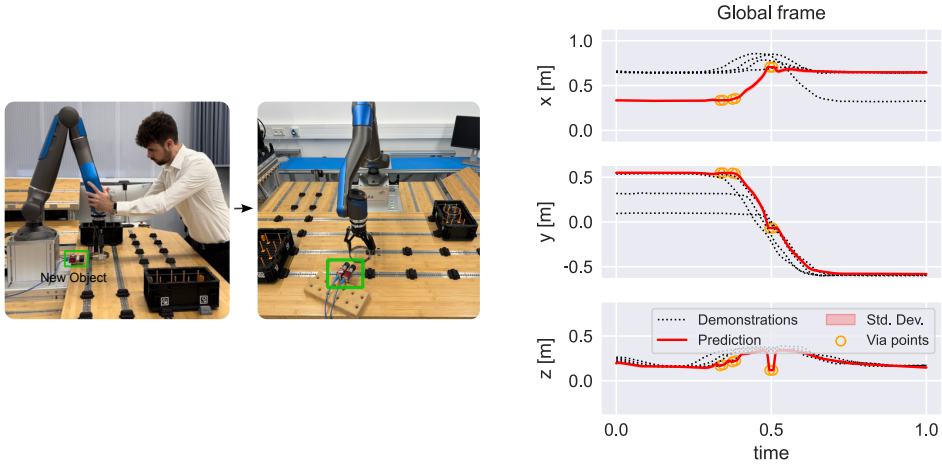


Figure 12: **Left:** Via-points with respect to the camera frame are generalized to different locations in the robot workspace, since they 'move' with the task parameters. **Right:** Trajectory distribution for a camera location different from Fig. 4, showing the generalization properties of our method.

A.4.7 Final model from Experiments 1–3

Figure 13 illustrates the expected robot motion after Experiments 1–3 (left) and the observed robot motion in terms of its Cartesian coordinates (right). The results demonstrate that the robot accurately follows the via-points, transitioning smoothly between them while adapting its precision to account for aleatoric uncertainty.

As the demonstrations are mapped to local frames, the robot does not need to align with these demonstrations in the global coordinate system. Instead, it is fulfilling the precision requirements of each frame model, extracted from (1), namely the new via-points. The robot remains stationary for $t > 1.0$, indicating a period of high compliance. At around $t > 1.2$, the robot begins to move in response to the emergence of via-points primarily along the z-axis, until it reaches the point where the camera checks the box.

Experiment 3 demonstrates that our approach allows for the compliant extension of a skill beyond its initial duration, which we argue is an important feature in LfD approaches. Particularly, since providing long demonstrations can be cumbersome, we believe that such an interactive and incremental framework is ideally suited to gradual and more intuitive skill transfer. It should be noted that just as a skill can be extended for $t > 1.0$, leveraging uncertainty-aware stiffness regulation and via-point definition also allows it to be extended for $t < 0.0$, such as to pick up a new object or change a tool before starting. A good practice for time-driven skills is to re-scale the input domain after via-points are added for unseen time instants, as this facilitates further skill extensions.

Overall, we believe that the capability to build on and refine existing skills, instead of re-demonstrating new skills each time even small variations occur, is a valuable feature for modern collaborative robots. The final execution of our experimental evaluation, as well as the steps leading up to it, described in Experiments 1–3, can be seen in detail in the accompanying video.

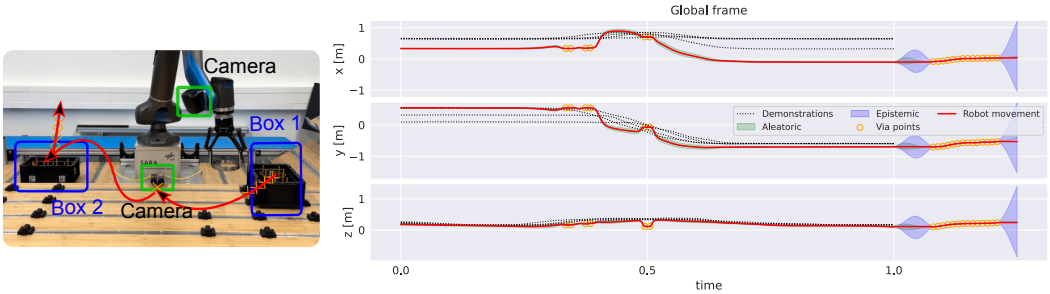


Figure 13: Final skill execution after Experiments 1–3. **Left:** Expected robot motion. The robot moves from box 1, passes in front of the newly introduced camera, moving to box 2 and finally moving upwards to verify the correct ring placement. Orange crosses indicate via-points. **Right:** Robot motion in Cartesian coordinates over time. The solid red lines shows the observed robot trajectories. Both aleatoric uncertainties are also depicted, as well as demonstrations and via-points.

A.4.8 Detailed stiffness profiles for Experiment 3 with hyperparameter sensitivity analysis

Analysis of the effect of l. Figures 14–15 give a detailed view of the stiffness profiles which we observed in Experiment 3 (we focus on the x dimension for simplicity). In Fig. 14 we see

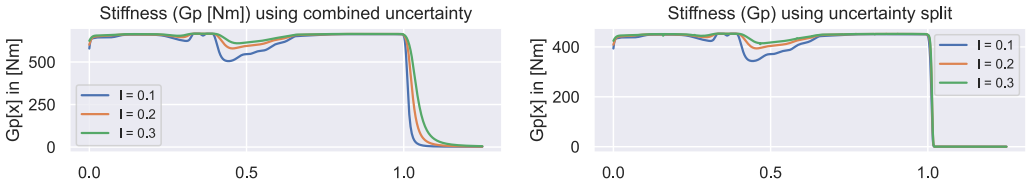


Figure 14: Stiffness values $\boldsymbol{G}_P$ for the x dimension, before via-points are added. **Left:** Stiffness computed using the original KMP covariance (3), through $\boldsymbol{G}_P = (\text{cov}\,[\boldsymbol{\xi}(\boldsymbol{s}^*)])^{-1}$. **Right:** Stiffness computed using our approach that separates epistemic and aleatoric uncertainties (5). The different line colors correspond to different kernel lenghts l, which we compared.

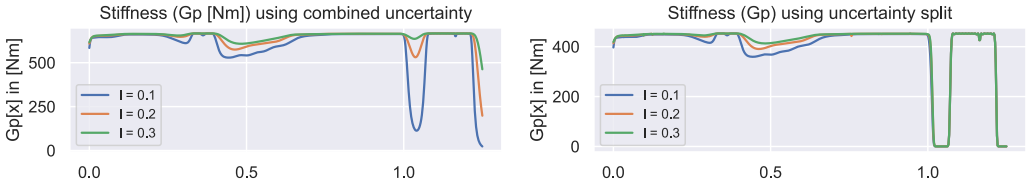


Figure 15: Stiffness values $\boldsymbol{G}_P$ for the x dimension, after via-points are added. Left and right plots show the curves obtained with uncertainties computed from (3) and (4), respectively, similarly to Fig. 14. Our approach (using uncertainty split) successfully makes the robot compliant in regions without data.

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 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 15 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 safety and efficiency.

Analysis of the effect of c_1, c_2. Figure 16 shows the stiffness profiles for changing values of c_1, c_2, using our approach (5) in the scenario with via-points after $t = 1.0$. In Fig. 16–left, we observe that increasing c_1 has the effect of more rapidly reaching minimum stiffness after $t = 1.0$. As discussed in 4.2, this is because it regulates how quickly the sigmoid function continuously switches between 0 and 1, which in our approach dictates the values of w_1, w_2. A more rapid switch is ensured by higher values of c_1. Figure 16–right highlights that smaller values of c_2 result in an earlier triggering of the transition. In summary, while c_1 regulates the rate of change, c_2 adjusts the set-point of the switching.

Note that, for certain choices of c_2 (in our experiments we used $c_2 = 1.5 \times 10^{-3}$), lower values of c_1 generate values of w_1 which are close to, but not exactly, 1.0. This deviation slightly reduces stiffness in regions influenced by aleatoric uncertainty (e.g., $t < 1.0$), as shown in Fig. 16–left. A similar effect is observed with c_2 (right), where lower c_2 values lead to increased stiffness in those regions. In our experiments we followed an approach of tuning c_1, c_2 based on the desired switching timing and adjusted δ_{al} and δ_{ep} to achieve the desired stiffness range for our robotic system.

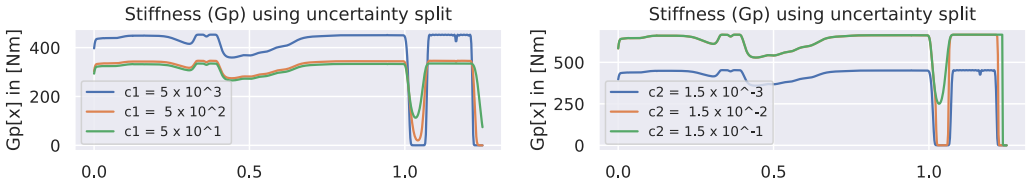


Figure 16: Stiffness values $\boldsymbol{G}_P$ for the x dimension, in the scenario where via-points are added after $t = 1.0$. The different line colors correspond to different sigmoid function parameters c_1 and c_2.

Analysis of the effect of δ_{al}, δ_{ep}. Finally, Fig. 17 shows the stiffness profiles for changing values of δ_{al}, δ_{ep}, using our approach (5) in the scenario with via-points. In Fig. 17–left we see that increasing δ_{al} has the effect of decreasing the overall stiffness in the regions where training data is present. This is due to δ_{al} enlarging the predicted covariance $\boldsymbol{\Sigma}^*_{\text{al}}$. Similarly, smaller values of δ_{ep} (e.g., $\delta_{\text{ep}} = 1.0$, which directly applies the epistemic uncertainty from (4) to compute stiffness) may not sufficiently amplify the uncertainty to ensure that stiffness approaches zero outside the training data (e.g. the time instant just after $t = 1.0$ in Fig. 7b–right). This reinforces our argument for introducing scaling parameters for the uncertainties, which are not tied to the KMP model's data fit but are instead focused on shaping the desired behavior of the system in leveraging these uncertainties.

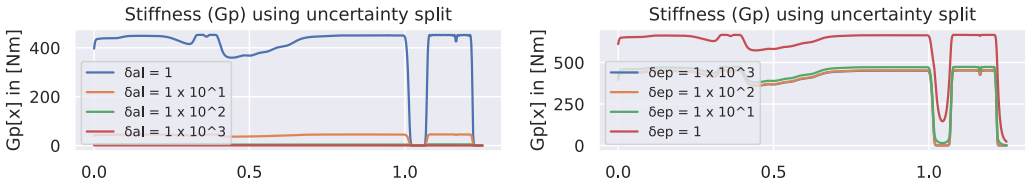
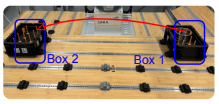
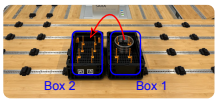
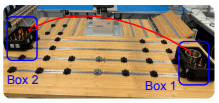
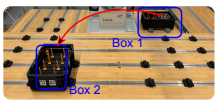
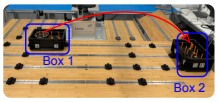
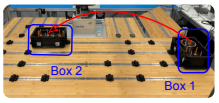
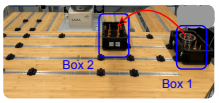
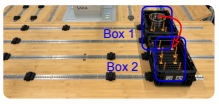
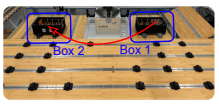
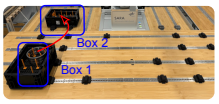
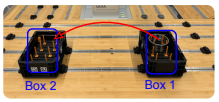
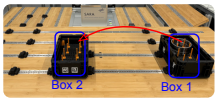
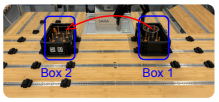
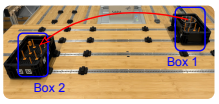
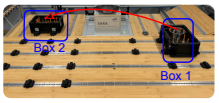


Figure 17: Stiffness values $\boldsymbol{G}_P$ for the x dimension, in the scenario where via-points are added after $t = 1.0$. The different line colors correspond to different epistemic and aleatoric scaling factors γ_{al} and γ_{al}.

Table 6: 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		✓	✓	✓
9		✗	✗	✓
10		✗	✗	✓
11		✓	✓	✓
12		✓	✓	✓
13		✓	✓	✓
14		✓	✓	✓
15		✗	✗	✓
	Success rate:	60 %	60 %	93.33 %