IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

Sequentially Teaching Sequential Tasks $(ST)^2$: Teaching Robots Long-horizon Manipulation Skills

Zlatan Ajanović*, Ravi Prakash*, Leandro de Souza Rosa*, Jens Kober

***Abstract*—Learning from demonstration has proved itself useful for teaching robots complex skills with high sample efficiency. However, teaching long-horizon tasks with multiple skills is challenging as deviations tend to accumulate, the distributional shift becomes more evident, and human teachers become fatigued over time, thereby increasing the likelihood of failure. To address these challenges, we introduce $(ST)^2$, a sequential method for learning long-horizon manipulation tasks that allows users to control the teaching flow by specifying key points, enabling structured and incremental demonstrations. Using this framework, we study how users respond to two teaching paradigms: (i) a traditional monolithic approach, in which users demonstrate the entire task trajectory at once, and (ii) a sequential approach, in which the task is segmented and demonstrated step by step. We conducted an extensive user study on the restocking task with 16 participants in a realistic retail store environment, evaluating the user preferences and effectiveness of the methods. User-level analysis showed superior performance for the sequential approach in most cases (10 users), compared with the monolithic approach (5 users), with one tie. Our subjective results indicate that some teachers prefer sequential teaching—as it allows them to teach complicated tasks iteratively—or others prefer teaching in one go due to its simplicity.**

***Index Terms*—Robot Learning, Interactive Imitation Learning, Long-horizon Manipulation.**

Fig. 1. Teaching a supermarket restocking task.

I. Introduction

Learning from Demonstrations (LfD) has emerged as an effective paradigm for teaching robots complex manipulation skills with minimal data and high sample efficiency. By leveraging expert demonstrations, robots can bypass low-level motion planning and focus on replicating high-level behaviors. However, when scaling to long-horizon tasks composed of multiple sequential or interdependent sub-tasks, LfD methods face significant challenges. Small deviations from the demonstration can accumulate over time, leading to cascading errors and distributional shifts that degrade policy performance.

Moreover, kinesthetic teaching (KT), the most common demonstration process due to its efficiency and performance [1], becomes increasingly demanding for the teacher as it is physically tiring for human instructors, particularly for long-horizon tasks, e.g., the supermarket restocking task illustrated in Figure 1. In this setup, the quality and consistency of demonstrations often decline as fatigue and the mental demand of thinking about several steps while providing demonstrations accumulate, deprecating the data quality. These limitations highlight the need for approaches that can structure, segment, and compose demonstrations in ways that support robust generalization over long task horizons.

Despite the progress on LfD and Interactive Imitation Learning (IIL) methods, the targeted tasks are usually simple and performed in a single motion [2], and extensions to long-horizon tasks have been tackled by several methods in the literature by ordering sub-tasks, or skills, to perform a complex task. As such, these methods assume a partition, or segmentation, of demonstrations into smaller tasks.

Towards assembling a long-horizon task from segmented demonstrations, early approaches [3] focus on sequencing sub-tasks using key-frames, which are important states shared among executions of the same task. Naturally, ordering non-sequential segments has also been explored, e.g., in [4], a low and a high-level policy are used to learn sub-tasks and their coordination, respectively. Recently, approaches aim at improving generalization, e.g., [5] uses demonstrations partitioned into an "alphabet" of sub-tasks, from which regular expressions are inferred to learn a program encompassing all demonstrations, making it possible to combine skills as a form of generalization to new tasks.

Alternatively, long-horizon tasks can be approached using Task and Motion Planning (TAMP) methods in synergy with LfD. As an example, [6] builds a likelihood model encoding the goal and teacher's intention, allowing to learn a task-tree

Z. Ajanovic is with the Computer Science Department, RWTH Aachen University, 52062 Aachen, Germany (e-mail: zlatan.ajanovic@ml.rwth-aachen.de).

R. Prakash is with Cyber Physical Systems, Indian Institute of Science Bengaluru, India (e-mail: ravipr@iisc.ac.in).

L. de Souza Rosa is with Alma Mater Studiorum Università di Bologna, Italy (e-mail: leandro.desouzarosa@unibo.it).

J. Kober is with the Cognitive Robotics Department, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: J.Kober@tudelft.nl).

* Equal contribution. Authors were affiliated with TU Delft at the time of the study.

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

used in planning new tasks by optimizing the co-occurrence of objects and the teacher's hands.

Regarding the data acquisition process, [7] frames long-horizon task learning with the TAMP framework, enabling the composition of complex tasks from simpler ones; it also allows for switching control to the human teacher when needed, and input information to be used for multiple sub-tasks, reducing the human-time requirements.

From the literature, it becomes clear that segmentation methods play a fundamental role in learning long-horizon tasks, since sub-tasks encoding the segments allow learning higher-level models easily by offloading the low-level training. This idea has been explored in [8], where pre-defined segments are identified on the demonstrations and used for learning sequential and hierarchical policies with reduced complexity.

To create segments from full demonstrations provided by the user, a common approach is to use Hidden Markov Models (HMMs)-based methods, an idea first introduced in [9], where a non-parametric model handles multi-scale segmentation due to its hierarchical auto-regressive nature. Bayesian inference has been used for segmentation and estimating their number automatically based on velocity profiles [10]. Furthermore, [11] considers the physical constraints of robotic manipulator trajectories to create an asymptotically stable dynamical system encoding the desired robot behavior. Nevertheless, these methods require the user to provide full demonstrations beforehand, without addressing the challenges that they impose on the human teachers.

When learning from a human teacher, there are two important aspects to consider. The first, and the focus of the literature mentioned above, is the learning efficiency due to the quality and scarcity of demonstrations that human teachers can produce and how they affect the policy learning [12]. The second are human aspects impacting the teaching, e.g., comfort, preferences, and safety, during the training sessions, which require user studies to evaluate.

For Human-Robot Collaboration (HRC) tasks, [13] evaluates the effects of task allocation, showing that operators reported greater satisfaction and production results improved whenever users could design their task sequence. Furthermore, [14] compares expert and novice teachers regarding their preferences for teamwork and effort, showing that expert users prioritize efficiency, while novice ones value comfort. Therefore, evaluating the user's preferences regarding the training framework is an important aspect in LfD and IIL.

In this paper, we focus on evaluating the human teacher's preferences when teaching long-horizon tasks in two frameworks: **i)** the traditional full task demonstration, hereafter named "monolithic"; **ii)** a sequential teaching method where teachers define segments according to their intuition.

Our key contributions are: 1) $(ST)^2$, a novel method for teaching long-horizon sequential manipulation tasks that allows the teacher to control teaching flow and teaching sub–tasks incrementally (sequentially). 2) A benchmark task with an analysis compatible with Failure Mode and Effects Analysis (FMEA) [15] (from the production engineering field), which allows for comparing both teaching frameworks directly. 3) An extensive user-study and analysis with 16 participants, yielding insights on human preferences in teaching and resulting policy quality.

II. Background

In this section, we provide the necessary background for our study, including the imitation learning approach adopted in this work, the kinesthetic teaching interface, and a problem formulation for long-horizon manipulation tasks.

A. Safe Interactive Movement Primitive Learning (SIMPLe)

The backbone method for our study is the Safe Interactive Movement Primitive Learning (SIMPLe) framework [16], which encodes full demonstrations using a time-dependent Gausian Processes (GPs)-based policy, providing a prediction of epistemic uncertainty that can be used for disturbance rejection or stiffness regulation [17].

In SIMPLe, n different demonstrations τ are recorded forming a dataset $\mathcal{D} = \{\tau_i\}_{i=1}^{n}$, each demonstration being composed of a sequence of M_i states $\tau_i = \{x_i^j\}_{j=1}^{M_i}$, which in turn represent the robot's state p (end-effector's Cartesian pose and grip state), and the sample's timestamp t, i.e., $x_i^j = (p_i^j, t_i^j)$. The labels for each sample are taken as the next point[1] in the same demonstration sequence, $y_i^j = x_i^{j+1}$.

The states and labels are concatenated into the $\mathcal{X}$ and $\mathcal{Y}$ sets, respectively, and a policy π is learned using a GP, providing goal predictions for the robot given its current x. Particularly, the goal is given by the GP's prediction mean and the variance provides an epistemic uncertainty estimation at the evaluation point, both computed as described in Equation (1), where the $k = k(x, x)$ is the current state's variance $x \notin \mathcal{X}$, $k_\star = k(\mathcal{X}, x)$ is the x and training inputs $\mathcal{X}$ variance, and $\mathcal{K} = \mathcal{K}(\mathcal{X}, \mathcal{X})$ is the covariance matrix of the $\mathcal{X}$ [18].

$$\begin{aligned} \mu(x) &= k_\star^\top \mathcal{K}^{-1} \mathcal{Y} \\ \sigma(x) &= k - k_\star^\top \mathcal{K}^{-1} k_\star \end{aligned} \tag{1}$$

A key aspect of SIMPLe is its time encoding in the GP's state, which evolves the policy as it progresses during task execution to provide superior encoding capabilities for complex trajectories, e.g., when similar states are visited multiple times, creating ambiguous estimations otherwise. To cope with the dynamic policies, SIMPLe employs the pseudo-GP kernel which correlates states only with their closest neighbors, as described in Equation (2), where $\tilde{k} = e^{-|p_i - p_j|\lambda^{-1}} e^{-|t_i - t_j|\lambda^{-1}}$ is a composed exponential kernel for the robot's pose p and time t.

$$k(x_i, x_j) = \begin{cases} 1, & \text{if } x_j = \arg\max_{x_k \in \mathcal{X}} \tilde{k}(x_i, x_k) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The correlation approximation in Equation (2) turns $\mathcal{K}$ into an identity matrix, avoiding the computationally expensive $\mathcal{K}^{-1}$ in Equation (1), which usually leads to prohibitively low control frequencies. Furthermore, when using k, the GP returns the closest point label as prediction, which can be

[1]The last point's label is itself.

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

interpreted as a graph, leading to the proposed name Graph Gaussian Process (GGP).

SIMPLe employs two mechanisms to ensure safe and smooth operation, which are not guaranteed given the GP approximation. The first being an adequate sampling rate ($>10\,\mathrm{Hz}$) to guarantee state coverage and the GGP's approximation stability. Second, it employs a Cartesian Impedance Controller (CIC) with dynamically regulated stiffness (K) and damping (D) matrices as described in Equation (3), where ρ is the actuation values of the robot's controller, and ∇J is the robot's Jacobian. K, D are dynamically computed using the current state's uncertainty $\sigma(x)$ as $K = \tilde{K}^{(1-\sigma(x))}/_{(1-\sigma_{th})}$, with σ_{th} being a variance threshold, $\tilde{K}$ a diagonal matrix with values limited to respect user-given maximum velocity and force thresholds, and $D = 2K^{1/2}$. As such, when uncertainty increases towards σ_{th}, the K automatically drops, stalling the robot's automatic operation and passing control back to the human teacher, who will provide feedback in the form of corrections used to refine the policy π.

$$\rho = \nabla J \left[K \left(\mu(x) - x \right) - D \left(\dot{p}_r \right) \right] \tag{3}$$

B. Teaching Interface and Corrections

The primary interface for providing demonstrations in this study is the kinesthetic teaching (KT), enabled by making the robot physically compliant, with its joints relaxed and easy to move, and in gravity compensation mode. From a control perspective, this is achieved by setting the robot's Cartesian Impedance Controller (CIC) to have very low K and D values, allowing the human teacher to grasp the robot's arm and guide its end-effector smoothly through the desired path. As such, KT allows non-expert users to easily teach robots new movements by guiding them without needing any programming knowledge.

As the user moves the robot, SIMPLe continuously records the robot's state x, including its spatial orientation and position (p) and the timestamp (t) at which that pose was reached, forming the demonstration trajectory τ which composes the states and labels ($\mathcal{X}$ and $\mathcal{Y}$) used to train the policy π.

Note that in our IIL setup, users provide $n = 1$ demonstrations to start up the policy, which can be iteratively refined through corrections. To do so, the robot starts performing the task controlled by the initial policy. If the robot deviates from the expected trajectory or stalls in regions of high uncertainty, the teacher can grasp the robot and move it to the desired pose. Such situations can be easily identified by the presence of external forces from the robot's joint-torque sensors, triggering SIMPLe to start recording the newly visited states x, which are appended to $\mathcal{X}$ and $\mathcal{Y}$, allowing for incorporating new information for the following run by simply retraining π. Furthermore, note that corrections are local, meaning that slightly displacing the robot to a well-defined region in the policy is sufficient to correct its policy, and the robot will automatically restart performing the task.

Besides KT, we allow users to provide high-level spoken language commands that steer the learning process and arbitrate autonomy between robot and human.

C. A Problem of Teaching Long-Horizon Manipulation Tasks

We consider the problem of teaching long-horizon manipulation tasks

$$\mathcal{T} = \{T_1, T_2, \ldots, T_m\},$$

where each sub-task T_i corresponds to achieving a sub-goal g_i (e.g., top grasping, collision-free reaching, placing, gripper release), and the overall execution of $\mathcal{T}$ achieves the final goal g_m. We assume that demonstrations are provided via KT, yet monolithic trajectories are prone to error accumulation and distributional shift, while also causing user fatigue. To mitigate this, the teacher should be able to dynamically control the teaching flow, switching between demonstration, autonomous execution, and correction, thereby enabling localized error recovery at the sub-goal level. Formally, this can be viewed as a simplified Semi-Markov Decision Process (SMDP) [19] with *options* representing sub-tasks $T_i \in \mathcal{T}$, where the challenge lies in learning consistent policies $o_i = (\pi_i, \beta_i)$ for achieving sub-goals $\{g_i\}$, while ensuring smooth transitions and maintaining usability for the human teacher.

III. Methodology

Our goal is to compare two IIL frameworks for teaching robotic manipulation tasks from the teacher's perspective. The first framework is the standard IIL one, hereinafter named "monolithic teaching", in which the user initially provides a demonstration of the entire task. In the second framework, users are allowed to demonstrate the task as a series of segments encoding "actions" by setting key-points on-the-fly and to control their creation and order with high-level user commands, thus enabling *progressive skill acquisition*, and *adaptive interaction* between human teachers and robots. In both cases, the user can provide real-time corrections during task execution, allowing for policy refinement.

To test the latter, we propose a novel method called $(ST)^2$, which implements a SIMPLe policy for each user-defined segment. The intuition behind $(ST)^2$'s user-centered segmentation is to balance instructional complexity with the teacher's capacity, mitigating cognitive overload while avoiding potential misalignment and ensuring the systematic buildup of skills.

A. Sequentially Teaching Sequential Tasks $(ST)^2$

Our teaching framework introduces a structured approach to robot learning through demonstration, featuring explicit control over when demonstrations are provided, policies are executed, and key-points are inserted (skills are segmented). This framework enables incremental task learning and supports fine-grained user intervention during the teaching process.

Since we consider the demonstration to be divided into s segments, the dataset becomes the set of demonstrations for each segment $\mathcal{D} = \{\tau_i\}_{i=1}^{s}$, with $\tau = \{x_i^j\}_{j=1}^{r_i}$, where r_i is the number of samples in the segment i. The segments are used to learn a SIMPLe policy for each segment in $\mathcal{D}$, forming the set $\Pi = \{\pi_i\}_{i=1}^{s}$.

Regarding the teaching flow, the teacher can perform five high-level commands to control the teaching process, summarized in Equation (4), giving the user control over task

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

segmentation by allowing the insertion of key-points and switching between demonstration and autonomous execution intuitively.

$$c \in \begin{cases} c_{\text{auto}}: & \text{executes full task autonomously} \\ c_{\text{demo}}: & \text{allows user to provide demonstration} \\ c_{\text{insertKP}}: & \text{inserts key-point and starts new segment} \\ c_{\text{next}}: & \text{executes action until the next key-point} \\ c_{\text{reset}}: & \text{resets to the episode's start} \end{cases} \tag{4}$$

Algorithm 1 outlines the sequential teaching framework designed for interactive and incremental policy learning. The process begins by initializing an empty dataset $\mathcal{D}$ and operating in demonstration gathering mode. At each time step, the system either records the robot's current pose (gathering data during demonstration) or autonomously executes the next action according to the current policy π_i (during autonomous execution). The teacher can dynamically influence the learning process through the high-level c commands: The command c_{demo} sets the system to gather a segment demonstration, and c_{insertKP} inserts a new key-point that ends the current segment demonstration and initializes a new segment demonstration. The command c_{reset} resets the environment and robot state to their initial conditions, and c_{next} runs the current policy until the subsequent key-point, and new points are added to the trajectory in case the user decides to correct the policy. After each iteration, policies are retrained on the updated dataset $\mathcal{D}$, ensuring continuous refinement as new data becomes available. For testing, c_{auto} sets the system to autonomous mode, in which the complete task is performed by sequentially executing the policies for each segment.

We highlight that the teacher is capable of overwriting segments by autonomously executing the previous actions and providing a new demonstration for the target segment and the subsequent ones, hence editing the segmentation's granularity. To do so, the user can autonomously execute the autonomous policy of segments that perform their respective sub-tasks successfully, and then take over the control to provide new demonstrations and key-points. Note that handling the transition to the newly presented demonstrations is unnecessary, since they start from the previous segment's ending point. To highlight, this feature differs from SIMPLe's local corrections, which can be used to fine-tune existing policies with kinesthetic feedback while executing autonomously.

As a final remark, note that SIMPLe's time encoding and graph-like correlation ensure that corrections will be close to the current evaluation point, and its safety mechanisms ensure smooth transitions between different segments. Therefore, no additional mechanisms are required to ensure smooth and safe trajectories.

Algorithm 1: Sequential Teaching Framework $(ST)^2$

Output: Updated dataset $\mathcal{D}$, set of learned policies $\Pi = \{\pi_i\}$

$\mathcal{D} \leftarrow \emptyset$ // Initialize dataset
$i \leftarrow 1$ // Initialize key-point index
$\tau_i \leftarrow \emptyset$ // Initialize segment data
$t \leftarrow 1$ // Initialize time
mode $\leftarrow$ c_{demo} // Start in demo mode
while *True* **do**
  $x \leftarrow$ [get_pose(), t]
  if mode $==$ demo **then**
    $\tau_i \leftarrow \tau_i \cup x$ // Append data
    $t \leftarrow t + 1$
    if c_{insertKP} **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \tau_i$ // Add segment to dataset
      $i \leftarrow i + 1$ // Insert new keypoint
      $t \leftarrow 1$
      $\tau_i \leftarrow \emptyset$ // Start new segment
  else if mode $==$ auto **then**
    if $p \approx p_i^r$ **then**
      mode $\leftarrow$ pause // Reached termination condition
    else
      execute($\pi_i(x)$)
      $t \leftarrow t + 1$
      if *user_input()* **then**
        $\tau_i \leftarrow \tau_i \cup x$
  if c_{demo} **then**
    mode $\leftarrow$ demo // Switch to demonstration
  else if c_{next} **then**
    mode $\leftarrow$ auto // Mode is autonomous
    $i \leftarrow i + 1$ // Move to next segment
    $t \leftarrow 1$
  else if c_{reset} **then**
    goto(p_1^1) // Reset to initial state
    $i \leftarrow 1$
    $t \leftarrow 1$
  $\Pi \leftarrow$ train($\mathcal{D}$) // Retrain policies

IV. Experiment

This section details the experimental setup designed to evaluate two imitation learning strategies (monolithic and sequential teaching) through a structured robotic manipulation task involving human participants.

A. Supermarket restocking task

For our user study, we have chosen the real-world supermarket restocking problem, in which a robot transfers a milk/yogurt carton from a storage box to a shelf, reflecting a practical scenario where robots assist in logistics and retail operations. This task presents several challenges that necessitate a multi-step approach due to the spatial constraints imposed by the shelf design, specifically the inability to place the carton in an upper shelf while holding it from the top. As the robot can grasp the carton from its box only from its top (due to the box sides), the robot must pick up the carton, place it on a table, and re-grasp it from the side before moving it to the shelf. One potential segmentation of this task into 10 sub-tasks

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

TABLE I
TASK BREAKDOWN FOR ROBOTIC MANIPULATION

Task ID	Description	Component
T1	Move to reach carton	arm
T2	Top grasp carton	gripper
T3	Move out of the box	arm
T4	Move above the location on the table	arm
T5	Release the grasp	gripper
T6	Reposition end-effector for side grasp	arm
T7	Side grasp carton	gripper
T8	Move to the shelf	arm
T9	Release the object	gripper
T10	Move out of the shelf	arm

is shown in Table I.

The supermarket restocking task was chosen for the user study as it embodies key characteristics essential for evaluating robotic manipulation and human teaching strategies. The task's moderate complexity ensures that it is neither trivial nor overwhelmingly difficult, requiring the robot to perform sequential, long-horizon actions such as re-grasping due to spatial constraints. This structure allows for clear observability and measurability, as success can be quantified through task completion time, grasp efficiency, and placement accuracy. Additionally, the task ensures safety and reproducibility, as it involves non-hazardous objects and can be consistently replicated across trials. Importantly, it offers high training value, providing a structured example of learning multi-step manipulation skills while maintaining a manageable cognitive load for human demonstrators.

Since we are interested in studying the differences in teaching a monolithic and segmented policy from the human teacher's perspective, the supermarket restocking task is adequate as it provides the necessary complexity to make these differences apparent. Nevertheless, other tasks can be considered for extensions of this study, reinforcing our findings or providing new insights.

B. Problem Complexity and Task Analysis

The supermarket restocking task has been selected through FMEA principles [15], a structured methodology for evaluating task complexity, identifying potential failure points, and assessing overall system performance, crucial aspects to IIL, where ensuring reliability is critical.

Regarding the task's complexity, we first decompose it to its minimum granularity, identifying individual steps involved in its execution, as summarized in Table I. For each step, Table II presents a summary of the identified potential failures, such as inaccuracies in motion replication, incorrect mental models, or hardware limitations, and their potential effects on performance, safety, and efficiency, enabling us to pinpoint critical vulnerabilities in both the learning and execution phases. Although the restocking problem might seem simplistic, it is clear that it is challenging, and many errors can occur in practice.

Regarding user performance, Section V presents results that evaluate task execution using objective and subjective metrics, user feedback, and teaching preferences.

As a final remark, perception is not considered in this study since the underlying learning method would require several demonstrations for learning generalized policies (e.g., [20]), greatly increasing the mental demand on the human teachers and the number of possible failure scenarios, hindering the user-preference analysis.

1) Monolithic Policy Teaching: In the monolithic policy teaching approach, the entire complex task is demonstrated to the robot as a single, uninterrupted movement. Demonstrating a task that involves multiple steps from start to finish in one go implies that the natural task breakdown into multiple steps must be executed fluently, which we extrapolate to be more difficult for inexperienced human teachers.

On the learning side, movements with small pose variations, e.g., during re-grasping, yield ambiguous points, ultimately leading to the learned police to either interpolate the movements to their average or falling into a deadlock. Even though this problem is mitigated by SIMPLe's time encoding, we argue that segmenting the task helps reduce potential ambiguous states created in these situations.

2) Sequential Policy Teaching: Besides reducing the policy learning complexity, the proposed segmentation has the advantage of keeping the semantic meaning intended by the users, which we extrapolate to be helpful to them, improving their capabilities of providing useful corrections and hence benefiting the learning process.

Nevertheless, a generic learning framework that considers segments would also require learning a full SMDP, a problem we avoid with the simplifications made in Section II-C, given that our goal is to evaluate the differences from the human teacher's perspective.

C. User Study

The user study[2] starts with participants reading and signing the provided information and informed consent sheets. The study is divided into three main sessions: Familiarization, Experimental, and Questionnaire.

During the *Familiarization Session* ($15\,\mathrm{min}$), participants receive instructions and feedback to familiarize themselves with the robot and KT setup, which is done by moving the robot in compliant mode to explore different joints, understanding the robot's joints limits, avoiding collisions, practicing opening and closing the gripper, and reorienting the gripper to pick objects from various orientations and positions. Additionally, participants practice teaching a simple pick-and-place sequence using both the monolithic and sequential methods.

In the *Experimental Session* ($30\,\mathrm{min}$), participants evaluate the two demonstration methods by manipulating the robot arm to perform a restocking task described in Section IV-A using the monolithic and $(ST)^2$ frameworks, in random order to avoid learning bias. Note that the task partition presented in Table I is not shown to the participants, who have total autonomy to segment the task according to their own understanding and preferences.

[2]Approved by the TU Delft Human Research Ethics Committee (HREC).

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

Fig. 2. Restocking task. A) Initial state, carton in the box, B) Intermediate placement for re-grasping, C) Final state, carton on the shelf.

TABLE II
ERRORS AND POTENTIAL EFFECTS IN ROBOTIC MANIPULATION FOR SUPERMARKET RESTOCKING

Category	Code	Error Name	Unwanted Effect
Teaching	ET01	Object moved by hand or falls	Missing object location at execution
	ET02	Object moved within the gripper	Missing object exact pose
	ET03	Moving close to other objects	Execution not robust, touching other objects
	ET04	Touching other objects while moving	Execution not robust, pushing the other objects
	ET05	Moving the arm while grasping or releasing	Grasping execution not robust
	ET06	Moving back and forth while demonstrating	Robot behavior not smooth
	ET07	Reaching joint limits while demonstrating	Robot behavior not smooth
	ET08	Non-perpendicular object placing	Placing execution not robust
	ET09	Non-perpendicular object placing grasping (e.g., pushing it while grasping)	Picking execution not robust
Task Planning	EP01	Grasping carton from the top and trying to place it directly (skipped steps)	Teaching failure
	EP02	Placing object at reachability limit (e.g., far placement, joint limits)	Impossible to continue the task
Segmenting	ES01	Not enough segments (far less than tasks)	$(ST)^2$ method not useful for corrections
	ES02	Meaningless key-frames	$(ST)^2$ method not useful for corrections
Correcting	EC01	Missing the desired time to start correction	Missed opportunity to correct
	EC02	Forgetting to use segmentation in followup demonstrations when correcting	$(ST)^2$ method not useful for correction
Setup	EG00	Different causes for gripper malfunctioning	Gripper does not perform taught action

Due to SIMPLe's encoding capabilities, a single initial demonstration and subsequent corrections are sufficient for learning a successful policy in most cases. All SIMPLe policies were trained in a few seconds on a standard laptop computer (with an `Intel Core 7` processor, without the use of a GPU) due to the GP's kernel approximation, rendering policy training time negligible in this study. Users can repeat the complete learning process from scratch up to 3 times for each method, capped by the elapsed time (1 h) and the user's level of fatigue.

During the *Questionnaire Session* (20 min), participants provide their opinions and preferences on each demonstration method by completing the NASA Task Load Index (NASA-TLX) questionnaire, a widely used subjective assessment tool to rate human perceived workload and performance for a given task. Participants also answer additional open-ended questions regarding their experiences and preferences for the demonstration methods.

Research questions: This study explores the following key questions:

- **RQ1:** Is the $(ST)^2$ (sequential) method more user-friendly or effective than the monolithic method?
- **RQ2:** Under what conditions does one teaching method become preferable over the other?

We hypothesize that users who successfully demonstrate the entire task in a single attempt may prefer the monolithic method due to its simplicity and speed. However, when the robot fails to reproduce the demonstrated behavior reliably, users may shift their preference toward the sequential approach, which offers greater control and opportunities for correction.

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

V. Results and Discussion

Our user study counts 16 participants, 3 female and 13 male, ranging from 22 to 40 years old. Among them, 14 had no experience with LfD, and 9 had never interacted with a robot before the study. A total of 6 participants were familiar with robotic manipulators. As such, the study is performed with users with diverse expertise levels, which is important since in LfD setups the learning performance is greatly impacted by the teacher's experience [12].

To illustrate the teaching flow and how autonomy switches between robot and user, while the latter provides demonstrations and segmentation key-points, we utilize the rooted branching tree representation exemplified in Figure 3. The top line (main) represents the complete task $\mathcal{T}$, with all sub-tasks, dots represent the corresponding keypoints at the end of the sub-tasks presented in Table I, and user-provided key-points are marked with a cross. Below the main line, the rooted branching tree represents all demonstrations in a single teaching trial; branches are created when the user requests autonomous execution until a certain point, and provides new demonstrations from that point onward. In this example, the user segments the demonstration with 7 keypoints (skipping T_3, T_4, and T_8). The user wanted to improve and requested autonomous execution until T_5, but gave up soon because keypoints were lacking, and the robot went too far from the expected trajectory, making the user request a reset. In the second correction run, the user executed autonomously until T_2 and continued providing demonstrations and key-points until the end. In some examples, users had more than 4 correction rounds, progressively advancing the success of sub-task execution.

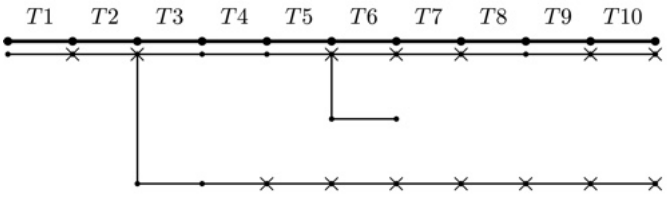


Fig. 3. User #03 $(ST)^2$ teaching flow.

A. User Preferences

Users were asked about their preference between teaching with the monolithic method or with $(ST)^2$ regarding their comfort, trust, expectations, and their perception of the robot's understanding of the task. Figure 4 summarizes the users' answers. While there is a clear distinction in their preference (i.e., they either prefer the monolithic or sequential method), most users declared being comfortable and trusting both methods, indicating that there is no downfall in teaching the robot in either way.

Furthermore, a significant portion of users favor $(ST)^2$ regarding trust, expectation, and their perception of the robot's understanding.

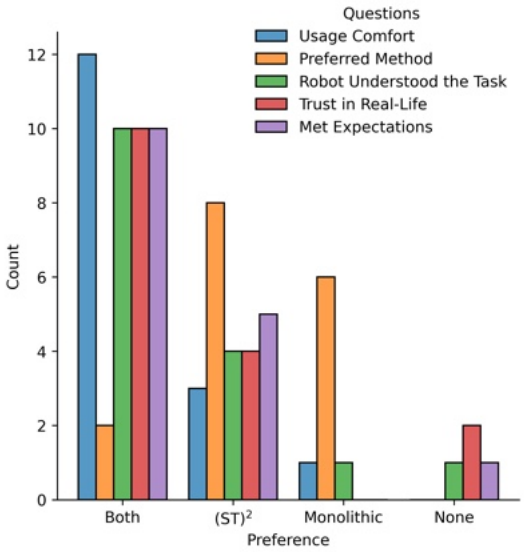


Fig. 4. Distribution of users' preferences and subjective evaluation regarding the learning methods.

B. Subjective User Evaluation

Figure 5 presents the NASA-TLX answers highlighting the users' subjective assessment of the tasks divided by the users' preferred method; 6, 8, and 2 participants preferred the monolithic, $(ST)^2$, and both methods, respectively. Results show a smaller physical demand using the monolithic method (overall $\mathrm{p-value} = 0.01$), which we attribute to its "in one-go" demonstration being less physically demanding, as pointed out by users.

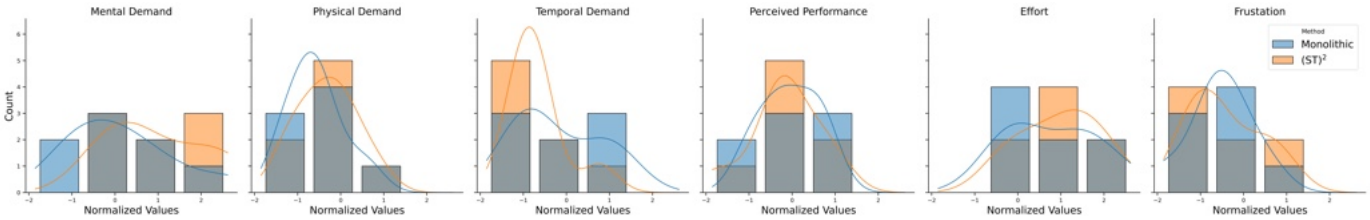
> *" It felt more satisfying getting the robot to perform the complete "choreography" correctly in a single movement. "*
> User #03 regarding the monolithic method.

Interestingly, $(ST)^2$ is perceived as the less temporally demanding method (overall $\mathrm{p-value} = 0.03$), which is caused by users investing a significant amount of time correcting the monolithic policy. Despite SIMPLe allowing for local corrections, finding the correct starting and stopping times for the corrective feedback proved challenging for non-expert users, who would continue to provide corrections until the task's end, as explicitly pointed out by five users in their comments.

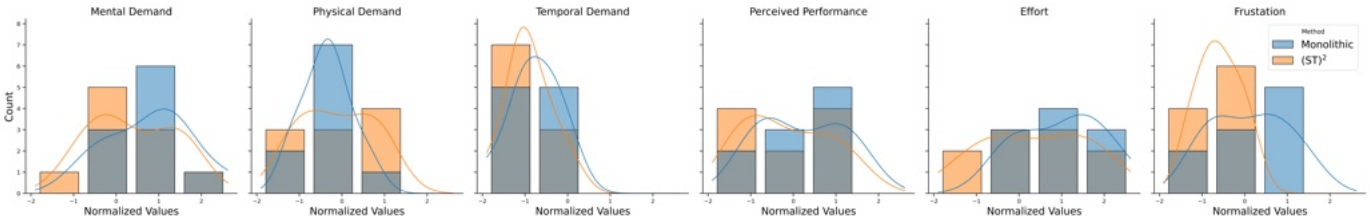
> *" Potentially, I would have to re-demonstrate everything to the robot if it fails. "*
> User #16 regarding the monolithic method.

An interesting point is that one could expect a higher mental demand when using $(ST)^2$, since users have to think about key-points. In fact, 3 users reported difficulties in deciding when to create a key-point. However, results show no significant difference in the mental demand (overall $\mathrm{p-value} = 0.42$). Upon further inspection, since $(ST)^2$ follows a natural

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.



(a) Users who preferred monolithic teaching.



(b) Users who preferred $(ST)^2$.

Fig. 5. Quantitative NASA-TLX workloads normalized (Z-score) per user. Smaller is better. Answers are divided according to the user's preferred method to highlight how the workloads affect their preferences. Users who preferred both methods are counted in both figures.

way of teaching tasks, the expected mental demand overhead was commonly overcome, as pointed out by 5 users.

> *" This task consists of sub-tasks. So I think I am more comfortable as a demonstrator to achieve the final goal if I can teach tasks separately. Also, I do not need to consider the optimization of the whole task, which reduces the need to think. "*
>
> User #02 regarding the $(ST)^2$ method.

> *" Yes, it allowed me to "explain" the task in segments, which is also how I prefer to do demonstrations to humans. Also, a small mistake felt less "stressful". "*
>
> User #07 regarding the $(ST)^2$ method.

Finally, we observe no statistical differences in the performance perceived by the users, nor in their effort and frustration self-evaluations.

C. Task Performance

Figure 6 presents the overall task scores for training the robot using both approaches and split by user preference. Despite a subtle alignment between users preferring $(ST)^2$ and their low score using the monolithic method, when evaluating all users altogether, we do not observe a significant score difference between methods ($\mathrm{p-value} = 0.98$), with success rates for the monolithic and $(ST)^2$ being 51.1 and 65.5, respectively. User-level analysis revealed that users improved performance with the sequential approach in most cases (10 users), compared with the monolithic approach (5 users), with one tie.

We speculate these results are related to the teaching flow difference between methods. Since the monolithic method requires users to time their corrections when the robot diverges

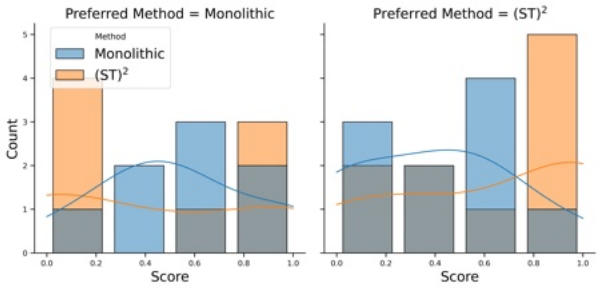


Fig. 6. Task performance score distribution measured $score = {}^{1}/_{\#t}$, where $\#t$ indicates number of necessary trials for a successful execution. Answers are divided according to the user's preferred method, and users who preferred both methods are counted in both figures. Higher is better.

from the expected path, users see it as a full demonstration, distracting them from the possibility of correcting locally. On the other hand, $(ST)^2$ allows for focusing corrections only on problematic segments, reinforcing the feeling of providing local corrections, as pointed out by 3 users.

> *" I prefer the $(ST)^2$ method because I was able to correct just after a certain step. If everything was good up to that point, I could "save my work". "*
>
> User #05 regarding the $(ST)^2$ method.

As a consequence, the focus on specific segments when using $(ST)^2$ allowed users to provide meaningful corrections only when needed, resulting in a successful task with fewer runs w.r.t. the monolithic method.

D. Quality Metrics

Table III provides a quality comparison between the trajectories generated by policies learned with both methods. The trajectory and torque jerk results show that $(ST)^2$ generates smoother trajectories, while also successfully performing the

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

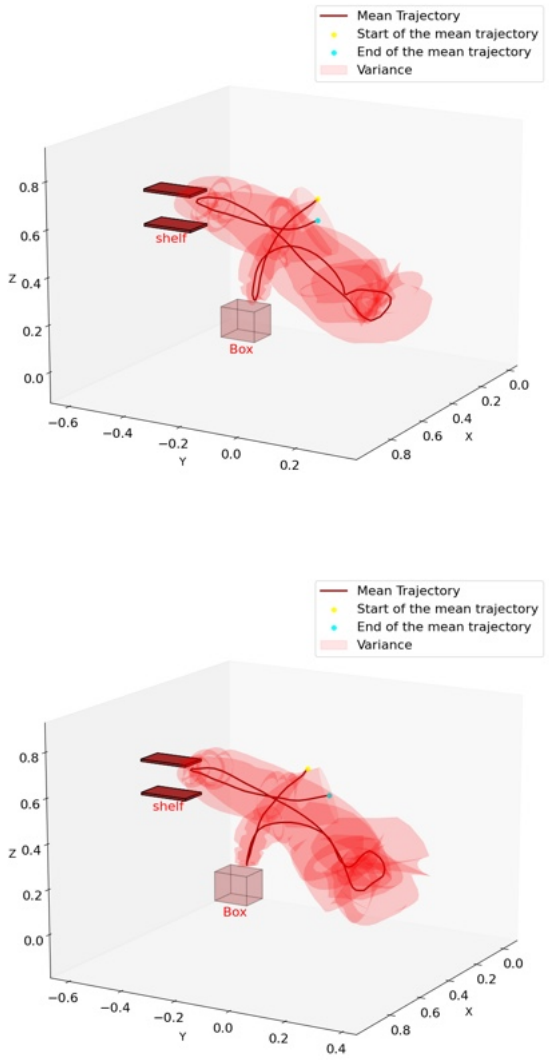


Fig. 7. Comparison of task learning methods: (top) Mean trajectory and variability across users for Monolithic Policy. (bottom) Mean trajectory and variability across demonstrations for $(ST)^2$ Policy.

task with similar speed but a higher success rate. Such results provide indicative evidence that our assumptions about the user being able to provide better demonstrations and corrections when he/she has control over the segmentation culminate in smoother and high-performing policies.

TABLE III
COMPARISON OF FROBENIUS NORMS OF JERK PRODUCT MATRICES (MEAN ± STD. DEV.) AND AVERAGE POLICY TIME TO COMPLETE THE TASK SUCCESSFULLY.

	Trajectory Jerk (10^4)	**Torque Jerk** (10^{10})	**Execution Time** (s)
Monolithic	10.5 ± 9.99	21.1 ± 32.5	46.1 ± 11.5
$(ST)^2$	8.91 ± 3.05	7.54 ± 4.11	46.3 ± 11.8

E. Learned Behavior

To illustrate the characteristics of the movement models learned using the monolithic and sequential policy teaching approaches, Figure 7 (top and bottom) shows the average trajectory and variance of the policies learned using the monolithic and $(ST)^2$ frameworks, providing a visualization of the trajectories' best estimate and their variability across different users. It can be observed that the learned behaviors from both teaching methods exhibit similar overall patterns, showing that the experiment design successfully achieved a task that provides participants with a Degrees of Freedom (DoFs) (segment choices) while limiting the variability between both frameworks, hence yielding a valid direct comparison.

F. Comparison Against Automatic Segmentation

Automatic segmentation methods rely on users providing a complete task demonstration before it can be segmented, usually using user-provided segmentation labels as a reference for evaluation. Note that in our setup, users create the segments on the fly (online), and the meaning of each segment is specific to each user, making a direct comparison against an automatically segmented trajectory invalid.

Furthermore, segmentation methods usually cluster similar points in the trajectory, without a guaranteed physical meaning to the human teacher. To exemplify this problem, Figure 8 presents an example trajectory segmented using [11], and the respective key-points created by the user (marked with a blue cross "x"), and the ending key-points of the sub-tasks identified in Table I by us (marked with a red diamond). Focusing on the automatically generated light-blue segment, one can observe that it contains points from when the robot was moving to fetch the milk carton and when it was placing it on the shelf, creating ambiguities for a policy trained on that segment. On the other hand, the user's segmentation often matches the task breakdown, highlighting how the user naturally segments the task with intrinsic semantic meaning and associated usefulness in the context of our teaching framework.

G. FMEA Errors

Finally, Table IV shows the frequencies of the possible errors highlighted in the FMEA analysis (Table II) that occurred during the experiments. The overall percentage of trials having errors is relatively large (76% for monolithic and 54.54% for $(ST)^2$), highlighting the complexity of the presented long-horizon task and its appropriateness for the study. The rate for errors related to multiple objects or sub-tasks is higher when using the monolithic method, indicating how focusing on a specific segment improves the teaching signal. Curiously, using $(ST)^2$ makes users reach the joint limits more often, which we speculate to be caused by disregarding the path taken by the robot to achieve the current pose between segments.

VI. CONCLUSION

This work presents a structured user study to evaluate human teachers' preferences regarding teaching long-horizon

IEEE Robotics & Automation Magazine (RAM) paper, presented at ICRA 2026, Vienna, Austria. Cite as RAM paper.

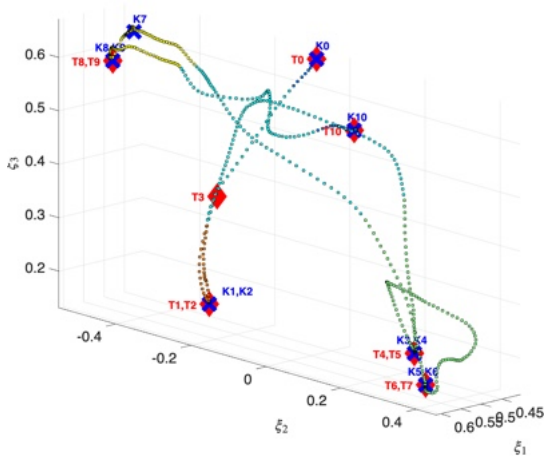


Fig. 8. Automatic segmentation of user's #16 demonstration using [11]. Segments (indicated by color) are defined by proximity. The user-defined key-frames (blue crosses annotated with K) often match the sub-tasks identified in Table I (indicated by diamonds annotated with T).

TABLE IV
FREQUENCY OF THE ERRORS SHOWN IN TABLE II MEASURED AS THE RATE BETWEEN THE NUMBER OF OCCURRENCES AND THE TOTAL NUMBER OF TRIALS (%). OMITTED ERRORS DID NOT OCCUR.

Error Code	ET01	ET02	ET03	ET04	ET05	ET06
Monolithic	16.27	0	13.95	18.60	6.97	4.65
$(ST)^2$	12.12	0	3.03	9.09	0	0
Error Code	ET07	ET08	ET09	EP01	EP02	EG00
Monolithic	2.32	6.97	0	4.65	2.32	37.2
$(ST)^2$	6.06	0	6.06	0	0	12.12

tasks as a monolithic policy or dividing the task into segments. To evaluate the former, we introduce a novel method called $(ST)^2$, in which teachers define segmentation points on-the-fly, allowing for progressive task learning, paired with a task designed for testing the methods while yielding consistent policies due to its constrained solution. Our evaluation indicates a reduction in temporal demand using the proposed method, since it allows users to focus only on the problematic parts of the policy. From the teaching strategies and preferences, the resulting insights show that some users prefer the proposed method because it provides better control over local correction, despite its increased temporal demand, while others lean towards the monolithic approach due to the simplicity of its single movement demonstration.

Several lessons emerged from our study. Notably, we observed a wide variability in user behavior and preferences, highlighting the importance of large and diverse participant groups to ensure statistically significant conclusions. Interestingly, user preference for a teaching method did not align with task success, similar for both methods, suggesting that perceived usability and actual performance may diverge.

Nevertheless, the policies created with $(ST)^2$ are shown to be smoother than the traditional monolithic approach, indicating that users provide better demonstrations and corrections over the segmented trajectories, without affecting the task execution time. User-level analysis revealed that the sequential approach improved performance in most cases compared with the monolithic approach.

Looking forward, several avenues for future work remain. First, we plan to explore alternative segmentation interfaces that offer users greater flexibility and control over how sub-tasks are defined. Second, we aim to extend our evaluation to a broader set of tasks with varying lengths and complexities to assess the generality of our approach. Finally, we are interested in studying long-term user behavior and co-learning. Specifically, how teaching efficiency and strategy evolve as users gain more experience with the system and teach multiple, diverse tasks.

ACKNOWLEDGEMENTS

The authors would like to thank Giovanni Franzese and Marta Ferraz for their valuable discussions and assistance at various stages of this work. LSR was sponsored within the SERICS - SEcurity and RIghts in the CyberSpace and received funding from the European Union Next-Generation EU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.3 – D.D. 1555 11/10/2022, PE00000013, and PE7 - CUP J33C22002810001, D.D. 341 15/03/2022, PE00000014). RP is sponsored by ARTPARK, IISc Bangalore. This project is made possible by a contribution from the National Growth Fund program NXTGEN Hightech. This manuscript reflects only the authors' views and opinions; neither the European Union nor the European Commission can be considered responsible for them. This work involved human subjects in its research. Approval of all ethical and experimental procedures and protocols was granted by TU Delft.

REFERENCES

[1] X. Jiang, P. Mattes, X. Jia, N. Schreiber, G. Neumann, and R. Lioutikov, "A comprehensive user study on augmented reality-based data collection interfaces for robot learning," in *19th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2024, pp. 333–342.

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

[3] C. Pérez-D'Arpino and J. A. Shah, "C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4058–4065.

[4] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "GTI: Learning to generalize across long-horizon tasks from human demonstrations," in *Robotics: Science and Systems*, 2020.

[5] N. Patton, K. Rahmani, M. Missula, J. Biswas, and I. Dillig, "Programming-by-demonstration for long-horizon robot tasks," *Proc. ACM Program. Lang.*, vol. 8, no. POPL, 2024.

[6] T. Welschehold, N. Abdo, C. Dornhege, and W. Burgard, "Combined task and action learning from human demonstrations for mobile manipulation applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4317–4324.

[7] A. Mandlekar, C. R. Garrett, D. Xu, and D. Fox, "Human-in-the-loop task and motion planning for imitation learning," in *7th Conference on Robot Learning*, 2023, pp. 3030–3060.

[8] B. Li, J. Li, T. Lu, Y. Cai, and S. Wang, "Hierarchical learning from demonstrations for long-horizon tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4545–4551.

[9] E. B. Fox, "Bayesian nonparametric learning of complex dynamical phenomena," Thesis, Massachusetts Institute of Technology, 2009, accepted: 2010-05-25T20:43:39Z.

[10] L. Gutzeit and F. Kirchner, "Unsupervised segmentation of human manipulation movements into building blocks," *IEEE Access*, vol. 10, pp. 125 723–125 734, 2022.

[11] N. Figueroa and A. Billard, "A physically-consistent bayesian non-parametric mixture model for dynamical system learning," in *2nd Conference on Robot Learning*, 2018, pp. 927–946.

[12] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *5th Conference on Robot Learning*, 2022, pp. 1678–1690.

[13] A. Tausch and A. Kluge, "The best task allocation process is to decide on one's own: effects of the allocation agent in human–robot interaction on perceived work characteristics and satisfaction," *Cogn. Technol. Work*, vol. 24, no. 1, p. 39–55, 2022.

[14] M. Pantano, A. Curioni, D. Regulin, T. Kamps, and D. Lee, "Effects of robotic expertise and task knowledge on physical ergonomics and joint efficiency in a human-robot collaboration task," in *IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, 2023, pp. 1–8.

[15] R. J. Mikulak, R. McDermott, and M. Beauregard, *The basics of FMEA*. CRC press, 2017.

[16] G. Franzese, L. de Souza Rosa, T. Verburg, L. Peternel, and J. Kober, "Interactive imitation learning of bimanual movement primitives," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 5, pp. 4006–4018, 2024.

[17] 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)*, 2021, pp. 7778–7785.

[18] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[19] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1, pp. 181–211, 1999.

[20] G. Franzese, R. Prakash, C. Della Santina, and J. Kober, "Generalizable motion policies through keypoint parameterization and transportation maps," *IEEE Transactions on Robotics*, vol. 41, pp. 4557–4573, 2025.