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

Screw Geometry Meets Bandits: Incremental Acquisition of Demonstrations to Generate Manipulation Plans

Dibyendu Das[1], Aditya Patankar[2], Nilanjan Chakraborty[2], C. R. Ramakrishnan[1], and I. V. Ramakrishnan[1]

***Abstract*— In this paper, we study the problem of methodically obtaining a sufficient set of kinesthetic demonstrations, one at a time, such that a robot can be confident of its ability to perform a complex manipulation task in a given region of its workspace. Although programming by demonstration has been an active area of research, the problems of checking whether a set of demonstrations is sufficient and systematically seeking additional demonstrations have remained open. We present an approach for the robot to incrementally and actively ask for new demonstration examples, one at a time, until the robot can assess with high confidence that it can perform the task successfully. Our approach uses (i) a screw geometric representation of motion to generate manipulation plans from demonstrations, which makes the sufficiency of a set of demonstrations *measurable*; (ii) a sampling strategy based on PAC-learning from multi-armed bandit optimization to evaluate the robot's ability to generate manipulation plans in a subregion of its task space; and (iii) a heuristic to seek additional demonstration from areas of weakness. We present results of a user study conducted with 22 participants (without any background in robotics) on two example manipulation tasks, namely pouring and scooping, to assess the utility and usability of our approach. The results show that a handful of examples (fewer than 10) were needed to successfully teach the robot to plan tasks. A short video supplement is available on YouTube: https://youtu.be/KbAPgIouIvo**

I. Introduction

The ability of robots to go beyond pick-and-place tasks and perform complex manipulation tasks is key to their use in many application areas, including flexible manufacturing, service robotics, and assistive robotics. In complex manipulation tasks, such as opening drawers, doors, pouring, scooping, etc., it is crucial for the motion of the end effector to follow geometric path constraints during task execution. It is often difficult to manually specify the complete set of constraints for each task.

One popular approach is to generate manipulation plans (a sequence of configurations of robot manipulators, such as its arms and grippers) directly based on a set of human-provided *demonstrations*. Such demonstrations can be obtained from video, teleoperation, kinesthetic manipulation of the robot's end effector, or simulations in virtual or augmented reality, e.g., see [1]–[11].

However, the following challenges hamper the use of current demonstration-based methods for practical applications: (a) the high cost of obtaining human-provided demonstrations, especially for kinesthetic or teleoperated demonstrations, which implies fewer demonstrations; (b) the opaqueness of the learning process – there is no way for the teacher to know which (if any) additional demonstrations would be useful. Thus, despite this rich history, the problem of evaluating the *sufficiency* of a given set of demonstrations and systematically *seeking* additional human-provided demonstrations has remained open.

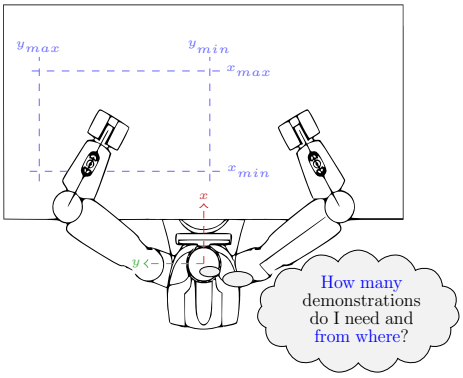


Fig. 1: Schematic sketch of a robot working in a table-top environment. The work area is indicated by the dashed rectangle.

A. Our Setting

We focus on manipulation tasks with rigid objects in a tabletop environment where all task-relevant objects are located within a *work area* (see Fig. 1). The work area lies entirely within the robot's reachable workspace. We assume that *positions* and *orientations* (i.e. *poses*) of task-relevant objects are known, possibly informed by vision and other sensor systems. Demonstrations are provided *kinesthetically*, where a human teacher holds the robot's end effector to perform the demonstrated task. Although resource-intensive, kinesthetic teaching is generally considered more usable and has a higher task success rate [1]. We restrict our attention to path-based task constraints (e.g. pulling a door by rotating it around its axis, holding a cup upright until it is rotated to pour out its contents into a bowl).

B. Our Approach

Preserving Task Constraints: Clearly, without an explicit specification of task constraints, it is impossible to determine whether a given manipulation plan satisfies the constraints. This encapsulates the fundamental difficulty in attaching any rigorous measure of *sufficiency* to a set of demonstrations. The key aspect of our approach that helps us circumvent this difficulty is that we generate manipulation plans from a kinesthetic demonstration via screw-linear interpolation, ScLERP [12], [13]. The manipulation plans generated by ScLERP combined with the Jacobian-pseudoinverse [14]

[1]Dept. of Computer Science, Stony Brook University, USA. {didas, cram, ram}@cs.stonybrook.edu.
[2]Dept. of Mech. Engg., Stony Brook University, USA. {aditya.patankar, nilanjan.chakraborty}@stonybrook.edu.

implicitly preserve the task-related constraints present in the demonstration (see §II). Since the satisfaction of task constraints is ensured, a demonstration is sufficient for a given task instance if a plan satisfying the joint limits of the manipulator can be generated. This gives us a concrete way to measure the sufficiency of a set of demonstrations.

Estimating Sufficiency: Let n denote the number of task-relevant objects. The pose of each object is an element of $SE(3)$, the group of rigid body configurations or motions. Let $\mathcal{X} \subseteq SE(3)^n$ be a set of possible poses of the n task-relevant objects. Each element of $\mathcal{X}$ is a *task instance*. Given $\mathcal{X}$, and a threshold $\beta \in [0, 1]$, we say that a set of demonstrations is sufficient if the probability that we can generate successful manipulation plans for task instances uniformly drawn from $\mathcal{X}$ exceeds β. Note that such a probability-based measure is, by itself, *coarse*: Although a set of demonstrations may be sufficient for $\mathcal{X}$, there may be regions within $\mathcal{X}$ where we cannot generate any successful plan. To identify such regions of weakness, we develop a procedure based on PAC ("Probably Approximately Correct") learning techniques from the "Multi-armed Bandit optimization" literature (see §III-D). PAC-learning in this setting lets us determine the minimum number of samples required in each region to obtain probability estimates with high confidence [15].

Incremental Demonstration Acquisition: A region with a low estimated probability of generating successful plans is a candidate for the next demonstration. With this idea at its core, we have built a system with a user interface that incrementally seeks demonstrations until $\mathcal{X}$ is covered, i.e., we can successfully generate manipulation plans with high probability within each subregion. The salient aspect of this system is that it permits non-experts to provide demonstrations. The effectiveness of the system in reducing the number of demonstrations was evaluated with a systematic user study with 22 non-expert participants providing demonstrations for two complex manipulation tasks, namely **scooping** and **pouring**. Thus, *we present an approach to build end-to-end systems with rigorous performance assurances that allows non-roboticists to program robots through intuitive kinesthetic demonstrations*.

C. Key Contributions of this Work

1) We define a concrete measure of *sufficiency* of a set of demonstrations from which we can generate manipulation plans for a given set of task instances (§III).
2) We provide a sampling procedure to estimate the sufficiency measure for subregions of the given set of task instances (§IV). This provides a basis for seeking additional demonstrations from the human teacher.
3) We present a systematic user study with *non-experts* to assess the utility of our approach in reducing the demonstration burden (§V).

The results of the user study show that, using our approach, fewer than 10 demonstrations are sufficient to generate successful plans over a specified work area compared to arbitrary acquisition of demonstrations (§V). We also performed simulation experiments to estimate distributions of the number of sufficient demonstrations for the two tasks (§VI).

We begin the technical development in this article with a more detailed description of the closely related work.

II. Related Work

Manipulation tasks, such as pouring from a container or opening a hinged door, involve path constraints on motion. Such constraints are hard and often impossible to specify generically with soundness or completeness guarantees. Learning from Demonstrations (LfD) [2], [16], [17] avoids this problem by using a set of demonstrations provided *a priori* to generate a constrained motion. This naturally raises two questions: 1) whether motion plans generated from demonstrations indeed satisfy the implicit constraints of the given task, and 2) whether a set of demonstrations are sufficient to generate robust motion plans for tasks in a specified work area.

A. Motion planning from demonstrations

There are a wide variety of techniques for generating motion plans from demonstrations, including probabilistic models based on hidden Markov models (HMM) [18], Gaussian mixture models (GMM) [19], diffusion models [20], [21], and bio-inspired techniques based on dynamical systems such as dynamical movement primitives (DMP) [22]–[25]. However, these works do not attempt to intentionally encode the task constraints that are present in complex manipulation tasks. Furthermore, any path constraints, if encoded, depend on the coordinate frame attached to the end effector, i.e., the constraint representation is not coordinate-invariant. Thus, these algorithms can fail due to changes in the poses of the task-relevant objects, and there is no formal way to establish whether a generated plan is correct in terms of ensuring the task-relevant constraints are satisfied.

Recent work has proposed the use of screw-geometric representation of constraints (which are coordinate-invariant by construction) in complex manipulation tasks [12], [13]. It has been shown that the *constraints, expressed as a sequence of constant screw motions, can be extracted from one demonstration* [13]. For new task instances, i.e., new poses of the task relevant objects, by suitably transferring the screw segments, and using screw linear interpolation (ScLERP) [13], [14] interleaved with Jacobian pseudo-inverse, one can generate joint-level motion plans that preserves path-based task constraints implicit in the demonstrations. By focusing on the parts of the trajectory close to the objects of interest, these works pay particular attention to the constraints most relevant to the task. Therefore, they are much more robust to variations in the poses of the task relevant objects. However, they may fail if the Jacobian pseudo-inverse leads to a joint-space path that hits joint limits. Nevertheless, *any feasible joint space plan (i.e., a plan satisfying the joint limits) generated using these approaches is guaranteed to satisfy the end-effector path constraints during motion*. Note that since, for tasks with end-effector constraints, it is much

easier to computationally check joint space feasibility of paths compared to task space feasibility, we use the algorithm proposed in [13] to plan motions.

B. Demonstration sufficiency using self-evaluation

The high-level objective of our work is to develop a framework in which the robot can *self-evaluate* its ability to generate low-level motion plans using a few demonstrations for complex manipulation tasks such as pouring, scooping, etc. This enables the robot to methodically seek additional demonstrations at targeted locations until it achieves a threshold of confidence in plan generation.

Previous works on self-evaluation of the satisfiability of task-related constraints [26], [27], have focused mainly on sensor information available from a perception system [28], [29] or high-level task specification [30]. However, none of the existing work on self-evaluation can be used to evaluate the robot's confidence in manipulation plan generation using kinesthetic demonstrations.

A recent work [31] explores *demonstration sufficiency* in the context of navigation tasks through *autonomous assessment*. While it focuses on demonstration phrased as expert-provided directions for navigating 2D grid environments, this approach does not directly extend to manipulation problems, where task constraints are embedded in the entire trajectory of the demonstration.

III. Self-Evaluation

The *pose of the end effector* refers to the position and orientation of the end effector frame – the reference frame fixed to the link of the robot arm beyond the last joint or on any object that is rigidly held by the robot. We consider manipulation tasks where the task specification includes the initial pose of the end-effector and the poses of other task-relevant objects that determine the end-effector's goal pose.

Task Instance: Let $\mathcal{X} \subset SE(3)^n$ be a compact set, where n is a positive integer representing the total number of task-relevant objects, including the object held by the robot. For a given manipulation task, an *instance of the task* is defined by $\mathbf{x} \in \mathcal{X}$. Thus, the set $\mathcal{X}$ defines the set of all task instances for a given task. We assume that all task-relevant objects are in the set of reachable poses of the robot's end effector.

A. Motion Planning from a Single Demonstration and Evaluation of Motion Plan

We will use the two-step motion planner discussed in [13] to generate motion plans for new task instances that differ from the demonstration instance. In particular, we will denote this motion plan generator by $\texttt{hasMotionPlan}(\mathbf{x},\ \mathbf{\Gamma})$ which takes as input any task instance $\mathbf{x}$, a demonstration $\mathbf{\Gamma}$ and decides whether or not a motion plan can be generated for $\mathbf{x}$ using $\mathbf{\Gamma}$. The planning process involves (a) transferring the guiding poses, $\mathbf{\Gamma}$, to a new set of guiding poses, $\mathbf{\Gamma}'$, to account for the new poses of the task-relevant objects and (b) using screw linear interpolation (ScLERP) interleaved with Jacobian pseudo-inverse to generate the joint-space path between two consecutive guiding poses. This motion planner ensures that the task space constraints in the demonstration represented as constant screw motion segments are always satisfied. However, the motion planner may fail to generate a plan for a given task instance due to a violation of joint limits. Therefore, *the planner used in this paper has the characteristic that when the planner returns a manipulation plan it is ensured that both task space and joint space constraints are satisfied*. Note that none of the other classes of planners for LfD, whether they be the more recent diffusion-based models or the classical DMP planners, can guarantee the satisfaction of path constraints in SE(3); making it difficult to formally (hence, computationally) characterize their performance.

Formally, we assume that given a demonstration $\mathbf{\Gamma}$ we have a mapping $f_{\mathbf{\Gamma}} : \mathcal{X} \to \{0, 1\}$, such that, for any new task instance $\mathbf{x} \in \mathcal{X}$, $f_{\mathbf{\Gamma}}(\mathbf{x}) = 1$, if and only if the robot can successfully generate a manipulation plan for the task instance $\mathbf{x}$ using the demonstration $\mathbf{\Gamma}$. Note that the algorithmic implementation of $f_{\mathbf{\Gamma}}(\mathbf{x})$ is $\texttt{hasMotionPlan}(\mathbf{x},\ \mathbf{\Gamma})$, defined above. Here, a plan is defined to be successful if it satisfies both task-space and joint-space constraints. If we consider a set of demonstrations $\mathcal{D} \equiv \{\mathbf{\Gamma}_i\}_{i=1}^n$ such that for any $\mathbf{x} \in \mathcal{X}$, $\max\limits_{\mathbf{\Gamma}_i} f_{\mathbf{\Gamma}_i}(\mathbf{x}) = 1$, then we have a set of demonstration examples such that the robot is guaranteed to generate a manipulation plan for any instance of a given task.

We formally characterize the sufficiency of a set of demonstrations $\mathcal{D}$ to cover a given set of task instances $\mathcal{X}$. If $\mathcal{D}$ is considered insufficient by this metric, we describe an approach to identify a region (subset) of $\mathcal{X}$ that contains a good candidate for a new demonstration. This forms the basis for the incremental acquisition of the demonstrations further developed in §IV.

B. Sufficiency of Demonstrations

For a demonstration, $\mathbf{\Gamma}_i$, let $\mathcal{B}(\mathbf{\Gamma}_i, \mathcal{X}) \subseteq \mathcal{X}$, be the set of task instances where $\mathbf{\Gamma}_i$ can be used to generate manipulation plans successfully. Thus,

$$\mathcal{B}\left(\mathbf{\Gamma}_i, \mathcal{X}\right) = \left\{\mathbf{x} \in \mathcal{X} \;\middle|\; f_{\mathbf{\Gamma}_i}\left(\mathbf{x}\right) = 1\right\} \tag{1}$$

Let $\mathrm{Vol}(A)$, with $A \subseteq \mathcal{X}$ be a volume measure [32] defined on $\mathcal{X}$.

Definition 1 (Coverage): The *coverage* of a demonstration $\mathbf{\Gamma}_i$ with respect to a set of task instances $\mathcal{X}$, denoted by $\mathbb{P}_{\mathcal{X}}(\mathbf{\Gamma}_i)$, is defined as:

$$\mathbb{P}_{\mathcal{X}}(\mathbf{\Gamma}_i) = \frac{\mathrm{Vol}(\mathcal{B}(\mathbf{\Gamma}_i, \mathcal{X}))}{\mathrm{Vol}(\mathcal{X})}$$

Let $\mathcal{D} \equiv \{\mathbf{\Gamma}_i\}_{i=1}^n$ be a set of demonstrations. We can lift $\mathcal{B}$ from individual demonstrations to sets of demonstrations. Let $\mathcal{B}(\mathcal{D}, \mathcal{X}) = \bigcup_{i=1}^n \mathcal{B}(\mathbf{\Gamma}_i, \mathcal{X})$. Then,

$$\mathcal{B}(\mathcal{D}, \mathcal{X}) = \left\{\mathbf{x} \in \mathcal{X} \;\middle|\; \max_{\mathbf{\Gamma}_i \in \mathcal{D}} f_{\mathbf{\Gamma}_i}(\mathbf{x}) = 1\right\} \tag{2}$$

Thus, $\mathcal{B}(\mathcal{D}, \mathcal{X})$ is the set of all task instances in $\mathcal{X}$ such that there is at least one demonstration $\mathcal{D}$ that can be used to generate a successful manipulation plan.

Analogously, we can lift the notion of *coverage* to sets of demonstrations:

$$\mathbb{P}_{\mathcal{X}}(\mathcal{D}) = \frac{\mathrm{Vol}(\mathcal{B}(\mathcal{D}, \mathcal{X}))}{\mathrm{Vol}(\mathcal{X})} \tag{3}$$

Definition 2 (Sufficiency): Given a probability threshold $0 \leq \beta \leq 1$, a set of demonstrations $\mathcal{D}$ is said to be *sufficient*

Algorithm 1 Obtaining a Sufficient Set of Demonstrations using `Self Evaluation`

Input: $K, \mathcal{X}_{1\ldots K}, \beta, \epsilon, \delta, \mathcal{D}_0$ ▷ $|\mathcal{D}_0| \geq 1$
Output: $\mathcal{D}$

1: $\mathcal{D} \leftarrow \mathcal{D}_0$
2: **while** true **do**
3: $j^*, \hat{\mu}_{j^*}, \mathcal{T}_{j^*} \leftarrow \texttt{getBestArm}(K, \epsilon, \delta, \mathcal{X}_{1\ldots K}, \mathcal{D})$
4: **if** $\hat{\mu}_{j^*} \leq 1 - \epsilon - \beta$ **then** ▷ see §IV-D
5: **break** ▷ End demonstration acquisition
6: $\mathbf{y}^* \leftarrow \texttt{selectFailedTaskIn}(\mathcal{T}_{j^*})$ ▷ see §IV-B
7: $\mathbf{\Gamma} \leftarrow \texttt{getNewDemonstrationAt}(\mathbf{y}^*)$ ▷ see §IV-C
8: $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathbf{\Gamma}\}$

for a set of task instances $\mathcal{X}$ with respect to β if $\mathbb{P}_{\mathcal{X}}(\mathcal{D}) \geq \beta$

C. Identification of New Demonstration Candidates

If a set of demonstrations $\mathcal{D}_0$ is insufficient for task instances $\mathcal{X}$, we would like to incrementally seek additional demonstrations, one at a time, thereby constructing a sequence of demonstration sets $\mathcal{D}_0, \mathcal{D}_1, \ldots$ such that some element of this sequence, say $\mathcal{D}_n$ is sufficient for $\mathcal{X}$.

We identify a small subset of task instances to seek the next demonstration by partitioning $\mathcal{X}$ into K disjoint compact sets $\mathcal{X}_j : 1 \leq j \leq K$. Note that

$$\mathbb{P}_{\mathcal{X}}(\mathcal{D}, \mathcal{X}) = \frac{\text{Vol}(\mathcal{B}(\mathcal{D}, \mathcal{X}))}{\text{Vol}(\mathcal{X})} = \frac{\sum_j \text{Vol}(\mathcal{B}(\mathcal{D}, \mathcal{X}_j))}{\sum_j \text{Vol}(\mathcal{X}_j)}$$

Then, if $\mathbb{P}_{\mathcal{X}}(\mathcal{D}_i) < \beta$, then there is a partition $\mathcal{X}_j$ such that $\mathbb{P}_{\mathcal{X}_j}(\mathcal{D}_i) < \beta$, that is, $\mathcal{D}_i$ is not sufficient for $\mathcal{X}_j$. Such a subset $\mathcal{X}_j$ contains candidate task instances for the next demonstration.

D. Formulation as Multi-Arm Bandit Optimization

We pose the problem of identifying the partition $\mathcal{X}_j$ that has the *least coverage,* $\mathbb{P}_{\mathcal{X}_j}(\mathcal{D}_i)$, in terms of the K-arm bandit optimization problem [33]. In the special case of Bernoulli K-arm bandit, each pull of the j-th arm yields a reward of 1 with an (unknown) probability p_j and 0 with probability $(1 - p_j)$. The optimization problem is to find the *best arm*, i.e., the one with the highest expected reward. Algorithms for solving this problem are compared based on the expected number of pulls (samples) needed to determine the best arm. For our problem, we create a K-arm bandit with partitions $\mathcal{X}_j$ comprising the K arms. Pulling the j-th arm corresponds to sampling a task instance $\mathbf{x}$ from $\mathcal{X}_j$. The reward 1 if $\mathbf{x} \notin \mathcal{B}(\mathcal{D}_i, \mathcal{X}_j)$, i.e. if the current set of demonstrations cannot generate a successful manipulation plan for $\mathbf{x}$; and 0 otherwise.

Note that in this formulation, the "best arm" corresponds to the partition that is least covered by $\mathcal{D}_i$. In other words, the best arm points to the partition where the robot has the least ability and hence should ask for the next demonstration. Once a new demonstration is obtained, the same process is repeated with the updated set of demonstrations $\mathcal{D}_{i+1}$. We stop when the current set of demonstrations is sufficient for all partitions.

IV. Incremental Demonstration Acquisition using Self-Evaluation

Algorithm 1 encodes our approach as pseudo-code to acquire demonstrations one at a time such that the set is sufficient for a given set of task instances $\mathcal{X}$. The algorithm takes the K partitions of task instances, $\mathcal{X}_{1\ldots K}$, the threshold β to determine sufficiency, and two parameters ϵ and δ, both $\in (0, 1)$, and a non-empty set of initial demonstrations, $\mathcal{D}_0$. Each element of the set $\mathcal{D}_0$ is a sequence of guiding poses, where two consecutive guiding poses constitute a constant screw motion. The algorithm proceeds by adding new demonstrations, initialized as $\mathcal{D}_0$. In each iteration,

(a) The algorithm first determines the partition j^* that is least covered by the current set of demonstrations (line 3) using bandit optimization as a subroutine (see §IV-A). Along with j^*, we also obtain $\hat{\mu}_{j^*}$, the *empirical* estimate of $1 - \mathbb{P}_{\mathcal{X}_{j^*}}$ for the current set of demonstrations.
(b) These sample task instances are used to select a suggested task instance $\mathbf{y}^*$ for the next demonstration (line 4); see §IV-B for details.
(c) Based on the suggested instance $\mathbf{y}^*$, we then obtain a new demonstration from the human teacher (line 5); see §IV-C for details.
(d) The new demonstration is added to the set of current demonstrations (line 6).

The collection of new demonstrations ends when we determine, with high confidence, that the current set of demonstrations is sufficient for every partition (see §IV-D).

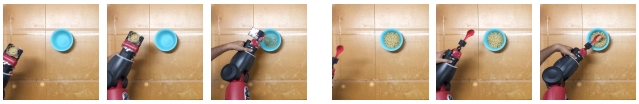
A. Finding an ϵ-Optimal Arm

Algorithm 2 uses the naïve (ϵ, δ)-PAC learning algorithm to identify an ϵ-optimal arm of our bandit formulation. In this algorithm, $\mathcal{T}_j^{\mathbf{1}}$ (line 4) is the set of all task instances for which we cannot find a motion plan and hence have reward 1. This algorithm ensures with confidence $1 - \delta$ that the empirical estimate of the expected reward for each arm, $\hat{\mu}_j$, is ϵ-close to the true expected reward μ_j. That is,

Algorithm 2 `getBestArm`

Input: $K, \epsilon, \delta, \mathcal{X}_{1\ldots K}, \mathcal{D}$
Output: $j^*, \hat{\mu}_{j^*}, \mathcal{T}_{j^*}^{\mathbf{1}}$

1: $N \leftarrow \frac{2}{\epsilon^2} \ln\left(\frac{2K}{\delta}\right)$ ▷ see § IV-A
2: **for** $j \in \{1, \cdots, K\}$ **do**
3: $\mathcal{T}_j \leftarrow \{\mathbf{x} \sim \mathcal{U}(\mathcal{X}_j)\}$ such that $|\mathcal{T}_j| = N$
4: $\mathcal{T}_j^{\mathbf{1}} \leftarrow \{\mathbf{x} \in \mathcal{T}_j \mid \neg\texttt{hasMotionPlan}(\mathbf{x}, \mathbf{\Gamma}) \quad \forall \mathbf{\Gamma} \in \mathcal{D}\}$
5: $\hat{\mu}_j \leftarrow |\mathcal{T}_j^{\mathbf{1}}| / |\mathcal{T}_j|$ ▷ Estimated expected reward
6: $j^* \leftarrow \arg\max_j \{\hat{\mu}_j\}$



(a) Demonstration of **pouring** (b) Demonstration of **scooping**

Fig. 2: Snapshots of kinesthetic demonstrations of **pouring** and **scooping** tasks. In each demonstration, the left frame is the initial pose and the right one is the final pose.

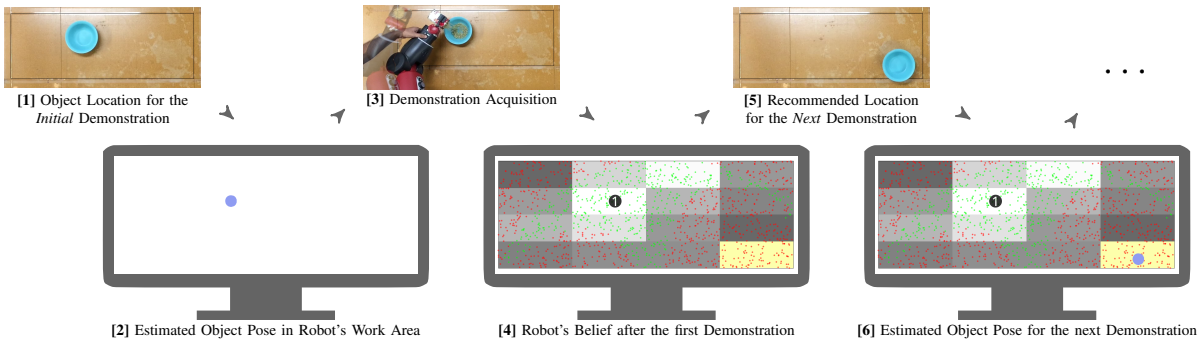


Fig. 3: Workflow of the demonstration acquisition process guided by the *self evaluation* algorithm. The upper row shows the events happening in the physical environment while the lower row shows the corresponding results on a user interface with the blue dot representing the estimated object pose.

$$\mathbb{P}\left(\max_j \{|\mu_j - \hat{\mu}_j|\} \leq \epsilon\right) \geq 1 - \delta.$$

The samples for each arm are drawn from uniformly distributed random task instances in each partition. Since each partition is a subset of $SE(3)$, we can use the techniques described in [34] used for sampling from $SE(3)$. The reward for each sample $\mathbf{x}$ is determined by checking if an executable motion plan can be derived for $\mathbf{x}$ based on the currently available set of demonstrations. We use the ScLERP based motion planner [13] to generate plans for each sample task instance. The ScLERP motion planner generates a set of guiding poses for each given task instance, but may fail to produce a sequence of feasible joint configurations needed to execute the plan even when such feasible configurations exist. This is a fundamental problem due to the inherent difficulty in always finding feasible paths in the joint space via inverse kinematics. Hence, determining rewards via ScLERP motion planner gives us an upper bound of rewards. Nevertheless, this only means that the set of demonstrations we find will be *conservative* — sufficient to cover the task instances — but may not be minimal. The minimum number of samples drawn from each arm, N, needed to ensure the confidence bound, is $\frac{2}{\epsilon^2}\ln\left(\frac{2K}{\delta}\right)$. To identify j^* such that μ_{j^*} is within ϵ of the actual maximum, we need to estimate the probabilities of the regions to within $\frac{\epsilon}{2}$ of their actuals in the worst case. When used with Hoeffding's inequality [35], this gives rise to the $\frac{2}{\epsilon^2}$ factor in the bound for N.

Note that we do not need to estimate each μ_j accurately to identify j^*; only the expected rewards of arms that are close to optimal need to be evaluated accurately. Several PAC-learning algorithms reduce the number of samples needed to identify an arm that is ϵ-close to optimal with confidence $(1-\delta)$ [36]. Other techniques such as UCB [37] may also be used to identify j^*. We are currently evaluating the suitability of such techniques and their effect on incremental demonstration acquisition.

B. Suggestion for the Next Demonstration

Having determined the partition with the least coverage, we seek the next demonstration using the following heuristic. Since we use ScLERP-based motion planner to check for feasible motion plans, we know that for each failed task instance (i.e. those in T_{j^*} with reward 1), there will be a screw segment $\left(\mathbf{g}^{(i)}, \mathbf{g}^{(i+1)}\right) \in \mathcal{G}$ such that $\mathbf{g}^{(i)}, \mathbf{g}^{(i+1)} \in \mathcal{G}$ for which we could not find a feasible joint-space path (due to joint limit violations). We choose the failed task instance $\mathbf{y}^*$ for which the joint-limit violation occurred in the earliest screw segment. Intuitively, a task failing early may have fewer viable alternative executions.

C. Obtaining a New Demonstration

A kinesthetic demonstration is collected, by recording a discrete sequence of joint angles $\mathbf{\Theta}$ and extracting from them a sequence of guiding poses $\mathbf{\Gamma}$ [13]. Note that even though we were seeking a demonstration for task instance $\mathbf{y}^*$, the task instance for the provided demonstration may be different – object poses may differ due to errors in manual placement.

D. Stopping Condition

This incremental process continues until the robot is confident enough that the overall success probability in the entire work area meets or exceeds a chosen threshold β i.e., $\mathbb{P}_{\mathcal{X}}(\mathcal{D}) \geq \beta$. In our setup μ_{j^*} is the (true) probability of failure to generate a successful motion plan in the region $\mathcal{X}_{j^*}$ corresponding to the optimal bandit arm j^*. When $\mu_{j^*} \leq 1-\beta$, we can then claim with confidence $1-\delta$ that for each partition $\mathcal{X}_j$, $\mathbb{P}_{\mathcal{X}_j}(\mathcal{D}_i) \geq \beta$. Note that we only know the empirical estimate $\hat{\mu}_j$ with the constraint $|\mu_j - \hat{\mu}_j| \leq \epsilon$. The *testable* condition used as the stopping condition (line 4 of Alg. 1) is $\hat{\mu}_{j^*} \leq 1-\epsilon-\beta$ since it implies $\mu_{j^*} \leq 1-\beta$. When this condition is satisfied, we can claim with confidence $(1-\delta)$ that the current set of demonstrations $\mathcal{D}$ is sufficient for all task instances in $\mathcal{X}$.

Note that although μ_{j^*} is monotonically non-increasing, we cannot bound the number of demonstrations needed to satisfy the stopping condition *a priori*. Alternatively, if we use a budget of a maximum number of demonstrations as our stopping criterion or stop early, we can compute the probability $\beta' = 1 - \epsilon - \hat{\mu}_{j^*}$ such that for each partition the robot believes it will be successful with probability β'. Here, $\hat{\mu}_{j^*}$ is the largest empirically estimated failure probability among all regions in the last iteration.

V. User Study Experiments and Results

To assess the effectiveness of incremental demonstration acquisition, we performed a systematic user study acquiring demonstrations in two kinds of trial: ***guided*** and ***unguided***. In a *guided* trial, an interface showed the positions of task-

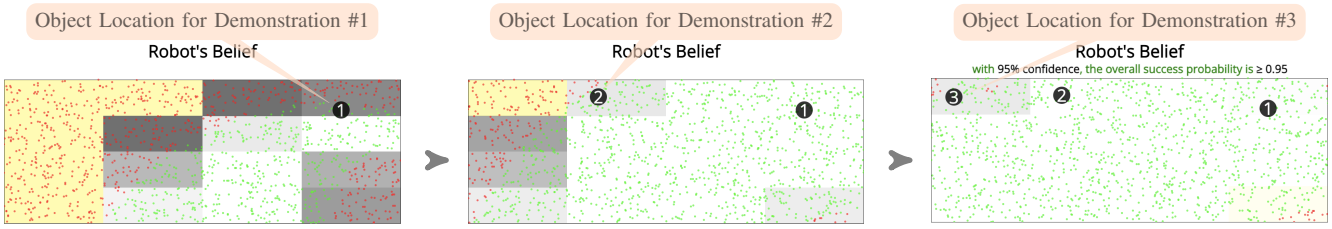


Fig. 4: Evolution of the robot's belief (shown as probability heat-map in varying shades of gray with white representing the success probability of 1.0) for one participant providing demonstrations of scooping using the guidance of *self evaluation* (Algorithm 1). Green and red dots represent locations of the task instances where plan generation succeeded and failed respectively. Yellow region is the recommendation provided by Algorithm 1 for obtaining the next demonstration.

relevant objects used in previous demonstrations (acquired using an object detection and localization system), a heatmap showing task instances for which motion planning using the given demonstrations succeeded or failed, and specifically identified candidate regions for the next demonstration (see §IV-B). In an unguided trial the users were only shown the positions of task-relevant objects in previous demonstrations, but not the robot's belief or a candidate for next demonstration. We used two tasks: ***pouring*** and ***scooping*** as examples of *complex manipulation tasks* (see Fig. 2). These tasks were chosen since they are characterized by constraints on the end-effector's motion that are key for their successful execution but difficult to specify beforehand, especially for non-experts. Moreover, the motion constraints in these two tasks differ significantly, making them well suited for the user study. Performing one after the other does not introduce any learning biases to the participants.

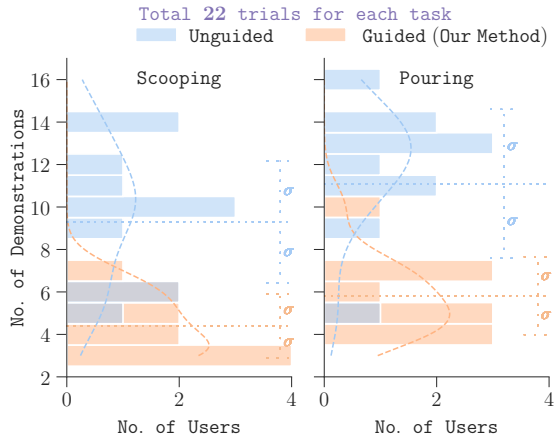


Fig. 5: Distribution of the number of demonstrations required to achieve the same objective of at least 95% overall success-rate with at least 95% confidence in Guided vs. Unguided trials.

Task	Number of Sufficient Demonstrations			
	Unguided		Guided (Our Method)	
	mean (μ)	variance (σ^2)	mean (μ)	variance (σ^2)
Scooping	9.7	9.42	4.5	2.07
Pouring	11.9	8.69	5.8	3.36

TABLE I: Number of *sufficient kinesthetic demonstrations* required to achieve a pre-determined success-rate of at least 95% for two complex manipulation tasks over 22 trials each.

A. User Experiment Setup and Participants

The experiments were carried out in a tabletop environment using the Baxter robot from ReThink Robotics using a rectangular work area of $0.38m \times 1.055m$, which is a significant fraction of the reachable area of the robot on the table (see Fig. 1). We used the Intel RealSense D415 RGB-D depth camera for vision sensing. The object's pose is defined as that of the 3D bounding box enclosing its corresponding 3D point cloud. Object detection is achieved using YOLOv5 [38], with detected pixels projected back to $\mathbb{R}^3$ using intrinsic camera parameters and depth values. Off-the-shelf algorithms are used for point-cloud processing and 3D bounding-box computation.

For the experiments, we chose $K = 16, \epsilon = 0.02, \beta = 0.95$, and $\delta = 0.05$ for Alg. 1. Thus, the objective of each participant was to provide a sufficient number of kinesthetic demonstrations for each task, so that, with 95% (i.e. $1 - \delta$) confidence, the robot could generate manipulation plans for at least 95% (i.e. β) of the task instances uniformly sampled from the work area. There was no limit on the amount of time or the number of demonstrations for each trial, i.e., each trial continued until the objective was achieved.

The study was carried out with 22 adult graduate students consisting of 5 female and 17 male, none of them having prior experience interacting with robots. The study lasted about 1.5 hours on average for each participant. Each participant underwent an initial training phase in which they were instructed about the tasks. They also moved the arm around in zero-gravity mode and the study commenced once the participants self-reported that they were comfortable in moving the arm anywhere in the workspace.

B. Experiment Design

Two types of trials (*guided* and *unguided*) applied to two tasks (*pouring* and *scooping*) lead to four experimental conditions: *guided pouring*, *unguided pouring*, *guided scooping*, and *unguided scooping*. To ensure that the trials were completed in a reasonable time, each participant was assigned to two different experimental conditions, always involving different tasks, while maintaining a fixed order: *the unguided condition always preceding the guided condition* – either (unguided pouring, guided scooping) or (unguided scooping, guided pouring). This design helps mitigate any potential bias introduced by the task order.

C. Experimental Findings

Since each of the 22 users was asked to participate in 2 (different tasks) of the 4 experimental conditions, for each experimental condition, we had a total of 11 trials. We now present our key experimental results.

1) Number of Kinesthetic Demonstrations

For both guided and unguided trials, each participant was able to teach the robot to achieve the given overall planning success rate of at least 95% with high confidence (at least 95%). The fact that each participant could teach the robot in guided and unguided trials provides empirical validation of the choice of the ScLERP-based planning approach. However, as shown in Table I and Fig. 5, for the unguided trials, the participants on average had to give more demonstrations. Using Alg. 1 helped reduce the number of demonstrations by 53.6% and 51.3% for the scooping and pouring tasks, respectively. Only 5 to 6 demonstrations were needed on average. Furthermore, as Table I shows, the variability between the participants was much smaller in the guided trial than in the unguided trial. Thus, *self evaluation* also ensures more uniformity in the teaching outcome of the participants despite potential individual differences in their ability to provide demonstrations.

Fig. 6 shows the variation in the overall success probability with the number of demonstrations. In unguided trials, although initially the success probability increased quickly, it soon plateaued, requiring significantly more demonstrations to reach the desired threshold of 0.95. Thus, without guidance, participants struggled to identify the remaining failure-prone areas in the workspace. However, with guidance, such a pronounced plateauing effect was not present.

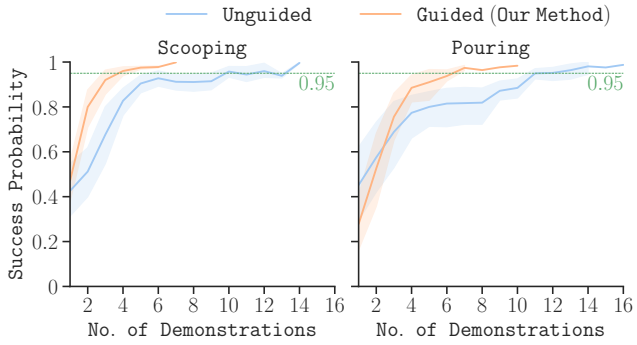


Fig. 6: Change of the Overall Success Probability with Incremental Acquisition of Kinesthetic Demonstrations. Thick lines represent the mean of all the 11 trials shown as the bands.

VI. Simulation Experiments and Results

The number of demonstrations obtained in a single trial in the user study may vary with different executions and also depends on the choice of K. To get an idea of the distribution of the number of sufficient demonstrations, we performed 1000 simulation trials, using self-evaluation, each for a different value of the number of disjoint regions K. The other parameters: ϵ, δ, and β were kept the same in all executions as before. Note that Alg. 1 is an interactive process in which after each iteration the robot asks for the next demonstration

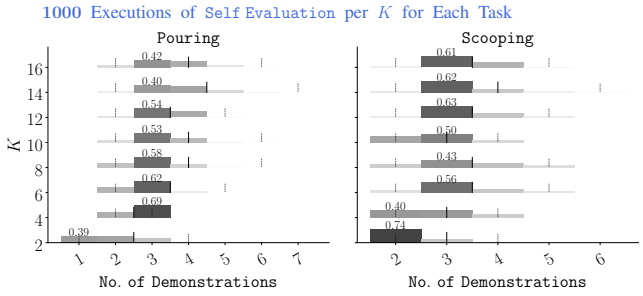
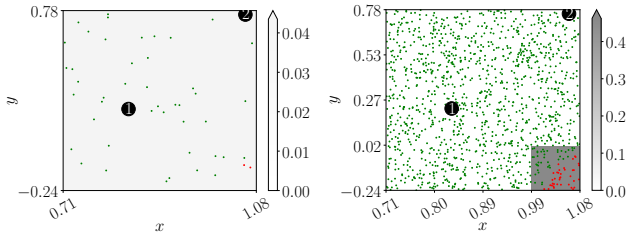


Fig. 7: Distribution (probability mass function) of the number of sufficient demonstrations for different values of the number of regions K. For each K, Algorithm 1 was executed 1000 times. The min, mean, mode and max are highlighted for each distribution.

from a human until it achieves its desired confidence. To bypass the human-robot interaction after each iteration in this simulation experiment, we pre-collected 32 demonstrations with at least one demonstration in any selected region. One of these pre-collected demonstrations was chosen in each iteration. The result of the simulation is shown in Fig. 7. The distributions show that we required a maximum of 7 and 6 demonstrations for pouring and scooping, respectively. Furthermore, in every simulation, for each value of K, the estimated expected number of sufficient demonstrations is always within ± 0.08 of the 99% confidence interval; and as K increases, the expected number of demonstrations increases only marginally (for example, for $K = 16$, it required only 3 to 4 demonstrations, on average, for both tasks). Note that smaller values of K require a smaller number of samples (from the bandit formulation). However, with smaller values of K, although we may achieve the desired confidence in the entire work area, there may be regions with much lower local success probabilities. This effect is explored in detail below.

For smaller values of K, the lower number of demonstrations (see Fig. 7) is due to the fact that we had fewer task samples to evaluate and the success guarantee is also not fine-grained, in the sense that although the overall probability guarantee may be satisfied, there may be regions where the performance is poor. We have demonstrated this with an example in Fig. 8. With $K = 1$ (left), an overall success probability of 0.95 is achieved in the entire work area using only 2 demonstrations. However, resampling the work area with the same demonstrations but with a higher value of $K = 16$ (right) shows that there exists a failure-prone region in the bottom right corner with a low individual success



(a) $K = 1$ (b) $K = 16$

Fig. 8: Effect of choosing two different values of the number of disjoint regions K, but with the same demonstrations.

probability of 0.6. The choice of K is application specific, so that the requirement of being able to succeed (with high probability) in the overall work area alone can be met by choosing a smaller value of K. However, to identify the pockets of failure and to succeed in each individual region, one may need to choose a higher K, which in turn comes with the additional cost of a larger number of samples, as well as a few more demonstrations.

VII. Conclusion and Future Work

This paper presents a novel approach to systematically obtain a sufficient set of kinesthetic demonstrations for complex manipulation tasks, one example at a time, such that the robot can develop a probabilistic confidence bound in its ability to generate feasible manipulation plans. Inspired by multi-arm bandit strategies, we propose an algorithm to partition the work area into disjoint subregions, ensuring successful plan generation in each region with a given probability that ultimately ensures overall success with high confidence.

The number of regions of interest for demonstrations (and the task space) increases exponentially with the number of task-relevant objects. Consequently, for the algorithm `getBestArm` (Alg. 2) to scale well, we will need to use a combination of non-naive sampling methods such as adaptive sampling [37], and hierarchical methods to focus on smaller sets of promising candidates.

Finally, we are studying whether demonstrations given in one context can be reused in a new environment, i.e., a new work area or robot. In this setting, it is possible to collect additional demonstrations in the new environment using the performance of the reused demonstrations as prior knowledge. Techniques for effectively reusing demonstrations in different environments remain an area to be further explored.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] S. Chernova and A. L. Thomaz, *Robot learning from human teachers*. Morgan & Claypool Publishers, 2014.

[3] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics & Automation Magazine*, pp. 44–54, 2010.

[4] K. Fischer, F. Kirstein, L. C. Jensen, N. Krüger, K. Kukliński, M. V. aus der Wieschen, and T. R. Savarimuthu, "A comparison of types of robot control for programming by demonstration," *HRI*, pp. 213–220, 2016.

[5] C. Pérez-D'Arpino and J. A. Shah, "C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy," in *ICRA*, 2017, pp. 4058–4065.

[6] E. G. Herrero, J. Ho, and O. Khatib, "Understanding and segmenting human demonstrations into reusable compliant primitives," in *IROS*, 2021.

[7] S. Bahl, A. Gupta, and D. Pathak, "Hierarchical neural dynamic policies," in *Robotics: Science and Systems*, 2021.

[8] T. Yu and Q. Chang, "User-guided motion planning with reinforcement learning for human-robot collaboration in smart manufacturing," *Expert Systems with Applications*, p. 118291, 2022.

[9] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *CoRL*, 2024.

[10] B. Zitkovich *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *CoRL*, 2023, pp. 2165–2183.

[11] L. Peternel and A. Ajoudani, "After a decade of teleimpedance: A survey," *IEEE Trans. Human-Machine System*, pp. 401–416, 2022.

[12] R. Laha, A. Rao, L. F. Figueredo, Q. Chang, S. Haddadin, and N. Chakraborty, "Point-to-point path planning based on user guidance and screw linear interpolation," in *Intl Design Engg. Tech, Conf. and Comp. and Info. in Engg. Conf*, vol. 85451. ASME, 2021.

[13] D. Mahalingam and N. Chakraborty, "Human-guided planning for complex manipulation tasks using the screw geometry of motion," in *ICRA*, 2023, pp. 7851–7857.

[14] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *IROS*, 2020, pp. 9480–9487.

[15] E. Even-Dar, S. Mannor, and Y. Mansour, "PAC bounds for multi-armed bandit and markov decision processes," in *CoLT*. Springer, 2002, pp. 255–270.

[16] A. G. Billard, S. Calinon, and R. Dillmann, "Learning from humans," *Springer handbook of robotics*, pp. 1995–2014, 2016.

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

[18] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *IJRR*, pp. 330–345, 2012.

[19] S. Calinon and A. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Advanced Robotics*, vol. 23, no. 15, pp. 2059–2076, 2009.

[20] C. Chi *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," *IJRR*, p. 02783649241273668, 2023.

[21] J. Urain *et al.*, "SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," in *ICRA*. IEEE, 2023, pp. 5923–5930.

[22] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.

[23] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *ICRA*, 2009, pp. 763–768.

[24] M. Saveriano, F. Franzel, and D. Lee, "Merging position and orientation motion primitives," in *ICRA*, 2019, pp. 7041–7047.

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

[26] R. Chatila, E. Renaudo *et al.*, "Toward self-aware robots," *Frontiers in Robotics and AI*, vol. 5, p. 88, 2018.

[27] A. Gautam, T. Whiting, X. Cao, M. A. Goodrich, and J. W. Crandall, "A method for designing autonomous robots that know their limits," in *ICRA*, 2022, pp. 121–127.

[28] M. Zillich, J. Prankl, T. Mörwald, and M. Vincze, "Knowing your limits-self-evaluation and prediction in object recognition," in *IROS*, 2011, pp. 813–820.

[29] K. N. Kaipa, A. S. Kankanhalli-Nagendra, and S. K. Gupta, "Toward estimating task execution confidence for robotic bin-picking applications," in *AAAI*, 2015.

[30] T. Frasca and M. Scheutz, "A framework for robot self-assessment of expected task performance," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 523–12 530, 2022.

[31] T. Trinh, H. Chen, and D. S. Brown, "Autonomous assessment of demonstration sufficiency via Bayesian inverse reinforcement learning," in *HRI*, 2024, pp. 725–733.

[32] G. S. Chirikjian and A. B. Kyatkin, *Harmonic analysis for engineers and applied scientists: updated and expanded edition*. Courier Dover Publications, 2016.

[33] A. Garivier and E. Kaufmann, "Optimal best arm identification with fixed confidence," in *CoLT*, 2016, pp. 998–1027.

[34] J. J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," in *IEEE ICRA*, vol. 4, 2004, pp. 3993–3998.

[35] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[36] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *Journal of Machine Learning Research*, vol. 5, no. Jun, pp. 623–648, 2004.

[37] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.

[38] G. Jocher *et al.*, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," *Zenodo*, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.3908559