Teaching the Teacher: Live Foundation Model and Augmented Reality Feedback for Human-to-Robot Skill Transfer

Nina Marie Moorman
nmoorman3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Matthew Luebbers*
Georgia Institute of Technology
Atlanta, Georgia, USA

Zhang Xi-Jia*
Georgia Institute of Technology
Atlanta, Georgia, USA

Yee Ching (Marcus) Lau
Georgia Institute of Technology
Atlanta, Georgia, USA

Yixing Yao
Georgia Institute of Technology
Atlanta, Georgia, USA

Megan Langwasser
Georgia Institute of Technology
Atlanta, Georgia, USA

Zulfiqar Zaidi
Georgia Institute of Technology
Atlanta, Georgia, USA

Letian Chen
Georgia Institute of Technology
Atlanta, Georgia, USA

Sanne van Waveren
Georgia Institute of Technology
Atlanta, Georgia, USA

Matthew Gombolay
Georgia Institute of Technology
Atlanta, Georgia, USA

Abstract

Deploying robots in dynamic, human-populated environments will require techniques for adaptable robot skill acquisition that extend beyond pre-programmed functionality. Learning from demonstration (LfD) methods enable robots to learn skills from human-provided trajectories demonstrated in situ. However, prior work has shown non-expert end-users struggle to provide demonstrations that enable robots to perform complex, multi-step tasks, or to generalize skill knowledge beyond a specific environment and task context. This work enables robots to actively participate in the situated learning interaction by autonomously providing bespoke guidance in response to end-users' demonstrations, thus improving end-users' ability to teach robots useful skills via LfD. We introduce a novel LfD system integrating foundation model (FM)-based textual feedback and augmented reality (AR)-based visual feedback. The FM and AR feedbacks operate synergistically, with FM feedback helping users break tasks down effectively and with AR feedback allowing users to quickly evaluate how well demonstrations perform and generalize. This system provides targeted, actionable guidance throughout the demonstration process: it enhances users' ability to define, decompose, and demonstrate modular, repurposable skills capable of accomplishing complex tasks. We validate our system with a human-subjects experiment in which participants receive bespoke feedback as they teach a robot via kinesthetic demonstrations in a pair of robotic manipulation domains. From this study, we observe positive results demonstrating that the combination of AR and FM feedback improves the quality and generalizability of robot policies, compared to AR feedback alone, FM feedback alone, or a baseline system where learned skills can be played physically on the robot.

CCS Concepts

• **Computer systems organization** → **External interfaces for robotics**; • **Human-centered computing** → *Mixed / augmented reality*; *Natural language interfaces.*

Keywords

learning from demonstration, foundation model, augmented reality

ACM Reference Format:
Nina Marie Moorman, Matthew Luebbers, Zhang Xi-Jia, Yee Ching (Marcus) Lau, Yixing Yao, Megan Langwasser, Zulfiqar Zaidi, Letian Chen, Sanne van Waveren, and Matthew Gombolay. 2026. Teaching the Teacher: Live Foundation Model and Augmented Reality Feedback for Human-to-Robot Skill Transfer. In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction (HRI '26), March 16–19, 2026, Edinburgh, Scotland, UK*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3757279.3788667

1 Introduction and Related Works

Learning from demonstration (LfD) methods enable robots to replicate behavior, given human-provided demonstration trajectories [1, 19]. While the objective of LfD is to enable non-expert end users to teach novel skills to robots without explicit programming, thus improving the flexibility and long-term maintainability of robot deployments in unstructured settings, in practice it is non-intuitive to teach robust skills via demonstration, especially for complex long-horizon tasks [8], such as preparing a meal.

Demonstrators often struggle to improve their demonstrations via trial-and-error [8], even with the ability to replay learned skills on the physical robot, which we denote as real robot replay (RRR) in this work [23]. As such, various instructional materials have been developed for demonstrators, including written and video-format

*Both authors contributed equally to this research.

HRI '26, Edinburgh, Scotland, UK

ACM ISBN 979-8-4007-2128-1/2026/03
https://doi.org/10.1145/3757279.3788667

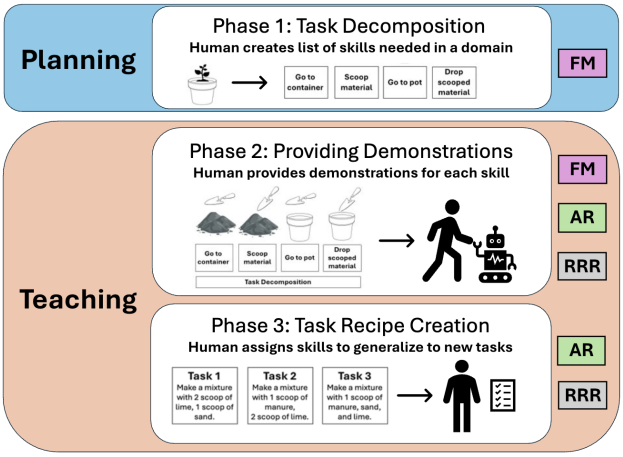


Figure 1: Our proposed system enables human demonstrators to teach a robot to accomplish general tasks within a domain across three phases. The human receives multiple modalities of bespoke feedback aimed at improving task decomposition and skill demonstration quality: Foundation Model (FM), Augmented Reality (AR), and Real Robot Replay (RRR) feedback.

user manuals [4]. One form of video-based instruction enables users to improve by practicing providing demonstrations in various domains, then comparing their demonstrations with those of a robotics expert [16]. However, none of these techniques are capable of giving targeted, actionable feedback in response to the human's demonstrations.

Our key insight is: to maximize the utility of LfD for non-expert end users, increased robot-to-human guidance and feedback is required both before and during the actual teaching process. We propose an interactive, situated learning [5] interface, where end users provide candidate skill decompositions and demonstrations in environmental context, which the robot uses to infer demonstrator intent; this intent is leveraged to provide tailored feedback on how the decompositions and demonstrations could be improved. We propose hierarchical, inter-related feedback including: (1) planning guidance to decide on a set of skills that will generalize to novel tasks and (2) teaching guidance to assess how provided skill demonstrations will perform and generalize.

To obtain planning feedback on whether a set of skills will generalize to novel tasks, we employ foundation models (FMs). FMs – models that are trained on large, broad sets of data to adapt to wide ranges of downstream tasks [2] – have the potential to serve as useful learning signals for demonstrators, as they can break down tasks into skills with varying degrees of abstraction [21, 28], an ability which is directly applicable to task decomposition.

Having received feedback about what skills to teach, the demonstrator now requires teaching feedback, i.e. guidance as to whether the learned skill will be executed as intended. We investigate providing FM teaching feedback on individual skill demonstrations, as FMs have been shown to be able to detect whether robot tasks were successful [7, 14]. To the authors' knowledge, prior work has yet to use FMs as a tool for giving feedback to improve LfD demonstrations.

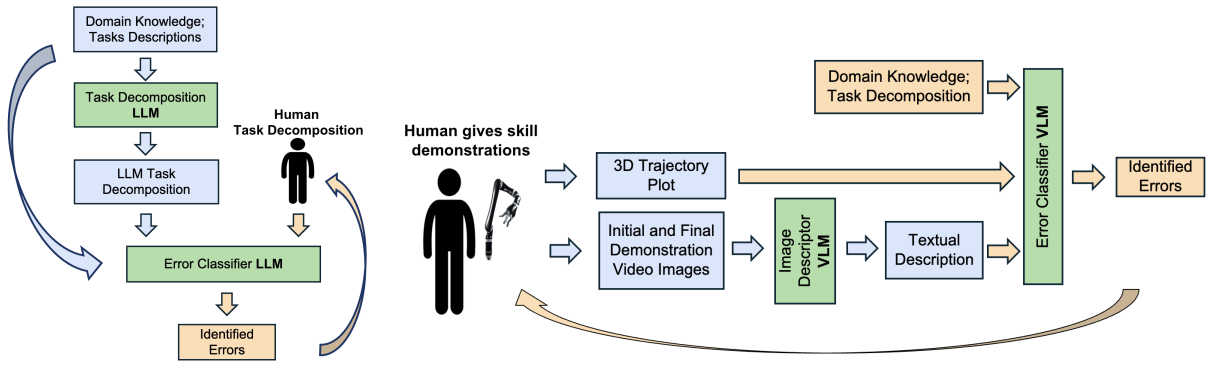
Augmented reality (AR) visualizations serve as an additional source of teaching feedback, as AR has the unique ability to project simulated robot behavior 3-dimensionally into the robot's environment, allowing users to see and understand what a robot is likely to do before deployment [20, 27]. AR visualizations have been used to analyze the behavior demonstrated to and learned via an LfD policy [13]. In our proposed system, AR visualization acts as a useful debugging tool for end users [11], allowing users to evaluate the safety of learned trajectories as well as whether the robot has learned skills as intended. Our work extends upon prior functionality, enabling users to visualize not only how a learned policy will execute, but also how it will generalize to changing tasks and environments.

Prior works have developed multi-modal human-robot interaction interfaces that incorporate AR and FM components [24]. For example, MARCER is a visual programming system that uses an FM to generate plans from pre-defined skills (i.e., users do not teach the skills) given user task descriptions; then, users can visualize and adjust the generated plans using AR [10]. Our system differs from prior approaches such as MARCER, in that ours supports task decomposition and physical skill teaching via human demonstration, enabling robots to learn new motor skills rather than relying on pre-defined ones. Thus, while MARCER is complementary to our approach, our system directly advances physical skill teaching and task decomposition for LfD.

We propose a novel LfD interactive feedback system where users program a full stack robot to perform multi-step tasks in which the robot automatically provides planning and teaching feedback. Our system (1) assists users in dividing a larger task into component base-level skills, to provide modularity and flexibility for new task variants, and (2) helps users ensure that the skills demonstrated are robust to changes in the environment or task description, allowing them to add or modify demonstrations to remedy any deficiencies. We empirically validate our system's benefits on demonstration quality and user experience through a human subjects experiment.

In this work, we contribute the following:

(1) We present a novel LfD interactive learning system enabling users to decompose and program complex multi-step robotic tasks with the aid of autonomously generated, bespoke AR and FM feedback. This feedback closes the interaction loop, improving demonstrator capability to effectively teach the robot.
(2) We empirically validate the benefit of our system in a human-subjects study spanning two manipulation domains, compared to ablated baselines: AR feedback alone, FM feedback alone, and a baseline system where learned skills can be played physically on the robot (RRR feedback).
(3) Our results demonstrate that the dual bespoke feedback modalities provided by our system lead to improved demonstration quality (in terms of measured robot performance on seen and unseen tasks), as well as enhanced perceptions of learned trust, and improved alignment between measured robot performance and user-predicted robot performance.



(a) FM planning feedback pipeline. **(b) FM teaching feedback pipeline.**

Figure 2: Overview of the foundation model feedback pipeline.

2 Methods

This section describes the different phases in which feedback is provided as well as the various feedback modes.

2.1 Feedback Phases

The automated feedback provided by our proposed system is spread across multiple phases, depicted in Fig. 1. Phase 1 (Task Decomposition) is a planning phase, where demonstrators define a list of low-level skills that must be taught to a robot for it to perform tasks in a chosen domain. This list of skills is then passed to the teaching phases: Phase 2 (Providing Skill Demonstrations) and Phase 3 (Task Recipe Creation). Here, demonstrators deliver demonstrations to the robot for each defined skill, and compose those skills into recipes for accomplishing novel tasks.

2.1.1 Phase 1: Task Decomposition. Given a set of tasks that must be accomplished in a domain (e.g., potting soil mixing), the first step of our LfD pipeline is for the demonstrator to decide what set of novel low-level skills (e.g., go to the sand bucket) are needed. Breaking down a complex task (e.g., create a soil mixture with two scoops of sand and one scoop of lime) into component skills is not intuitive [8]. Skills must strike the right balance of abstraction, with enough specificity to successfully learn the skill, but enough generalizability to be applicable to multiple tasks, and be robust to changes in environment configuration. Providing feedback in this phase, which is prior to the user taking time to physically demonstrate each skill, can save the demonstrator time and effort, thereby improving the experience and efficiency of teaching.

2.1.2 Phase 2: Providing Skill Demonstrations. Once the demonstrator has decided on a list of skills to teach the robot, the demonstrator proceeds to provide demonstrations for each skill. Collecting demonstrations can be done in many ways for LfD [19]; for our implementation, we chose kinesthetic teaching (i.e., physically moving the robot through each skill and recording the trajectories). Providing feedback in this phase enables the user to quickly identify errors and re-record demonstrations as needed.

2.1.3 Phase 3: Task Recipe Creation. Once all skills have been demonstrated, the user can compose these skills sequentially to enable the robot to execute novel tasks. Providing feedback in this phase enables the demonstrator to observe how their skills chain together to execute a multi-step task, allowing them to add, redefine, or alter skills as needed.

2.2 Feedback Modes

Our final contributed system consists of FM Planning feedback (Section 2.2.1) for task decomposition and AR Teaching feedback for skill learning (Section 2.2.2). We include Real Robot Replay (Section 2.2.3) as a baseline condition rather than part of the system itself. We also evaluated an FM Teaching feedback module (Section 2.2.1) for skill learning, but because it did not improve teaching effectiveness, it is treated as an ablation and is not part of the final system. We employ Chatgpt-4o as our foundation model throughout all phases; full prompts and inputs (described in the following sections) and example FM outputs are provided in Section E of the Appendix.

2.2.1 Foundation Model Feedback. Two forms of FM feedback are evaluated in this work.

FM Planning Feedback The large language model (LLM) prompting process for providing feedback to the demonstrator-provided task decomposition (Section 2.1.1) is depicted in Fig. 2a. First, the Task Decomposition LLM generates a candidate task decomposition using high-level environment and domain descriptions and list of tasks as input. Next, the Error Classifier LLM uses the Task Decomposition LLM's candidate task decomposition as a reference to provide feedback for a user's input. The Error Classifier LLM is prompted to direct its feedback according to a predefined list of errors, defined in Section E of the Appendix. The two-step LLM prompting architecture is informed by the general technique employed in [15] who show that passing the initial response of an LLM into its subsequent prompt is helpful for improving the quality of responses.

FM Teaching Feedback The vision language model (VLM) prompting process for providing personalized feedback to the demonstrator on the quality of their skill demonstrations (Section 2.1.2) is depicted in Fig. 2b. First, the Image Descriptor VLM summarizes the initial and final images of the demonstration's video recording.

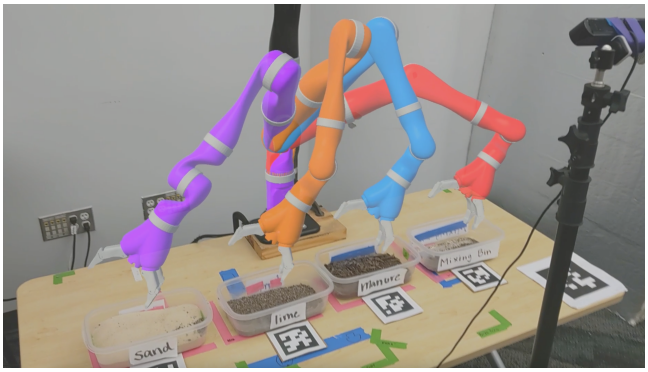


Figure 3: AR generalization capability for the soil mixing domain, with overlaid AR feedback showing an animation of a learned skill, applied relative to multiple bins as separate robot animations.

Next, we input these summaries into the Error Classifier VLM to provide personalized feedback to the demonstrator on the quality of their skill demonstrations. While these summaries are informative regarding the state change resulting from the demonstration, they do not improve the VLM's spacial and temporal understanding of the robot's trajectory. To address this gap, we take inspiration from prior visual prompting work that overlays a trajectory on an image [22] and propose a novel visual prompt component: a 3D plot, colored according to a temporal gradient, of the demonstration's end effector trajectory from the same point of view as the image data. The Error Classifier VLM is prompted to direct its feedback according to a predfined list of errors, defined in Section E of the Appendix. Although we evaluated FM Teaching feedback in our human-subjects study, it did not improve participants' ability to teach skills. As such, FM Teaching feedback is not a part of our final, contributed system, which consists only of FM Planning and AR Teaching feedback. We include the module's description and its negative result solely for completeness and reproducibility.

2.2.2 Augmented Reality Feedback. Throughout the teaching phases, users will be able to view simulated robot trajectories projected into and animated within the environment using an augmented reality headset (Microsoft HoloLens 2). For each AR feedback type, users are able to view the animated robot from any angle, as well as play, pause, and alter the playback speed of the animation.

AR Teaching Feedback

(1) Learned Skill Visualization: Users can view skill trajectories learned from their demonstrations, similar to [13], but instead of showing static trajectories, the entire robot is shown animated. This feedback enables users to ascertain whether the robot is likely to perform as intended.
(2) Skill Generalizability Visualization: Users can view skill trajectories learned from their provided demonstrations, applied to one or more (up to four) objects of interest simultaneously. For example, a single learned "scoop" skill could be visualized with respect to bins of sand, lime, and manure. To visualize this, multiple robots of different colors are animated simultaneously (Figure 3.) This feedback enables users to see how their skills generalize to different setups.
(3) Learned Task Visualization: Having specified a recipe of learned skills to accomplish a task, users can view the learned task in AR, animated end-to-end. This feedback enables rapid debugging of taught skills, without requiring time-consuming and dangerous testing on the real robot.

2.2.3 Real Robot Replay Feedback. As a baseline feedback modality, participants could replay learned skills on the physical robot, which is a de facto standard debugging mechanism in LfD systems. Users were able to deploy either individual learned skills or full task recipes on the robot.

2.3 Robot Learning Algorithm

In this work, we only collect one demonstration from each participant per skill. We employ a trajectory-based robot learning from demonstration algorithm, namely Cartesian Dynamic Movement Primitives[1] (DMP), as they have the ability to learn quickly (providing more time for human-in-the-loop trial-and-error) from limited demonstration data, without need for a robust environment simulator, while possessing sufficient generalizability to handle minor modifications in environment or task specification (e.g., changing object locations). The state space includes the robot joint angles, gripper state (open or closed), and the position and orientation of interactable objects in the environment with respect to the robot base link. When deploying the learned skill we fit the Cartesian-space DMP to the participant's demonstration's end-effector trajectory, adapted to the desired start and goal location. We then use inverse kinematics to obtain a joint-space waypoint sequence. The gripper's actuation state (open/closed) is not learned using a DMP - instead it is resampled with nearest-neighbor interpolation for the duration of the learned skill.

The desired starting location of a skill is defined as the current position of the arm. To enable skill generalizability to multiple items and item locations, we allow participants to specify a skill's goal location with respect to any object whose location is tracked using AprilTags. For instance, say a participant teaches the skill "go to bucket" and ends their demonstration 5 inches above the sand bucket. If the participant specifies that they are teaching that skill with respect to the sand bucket, then the system learns that the goal location should be 5 inches above the assigned object for that skill. If the sand bucket later moves, or if the participant wants to apply this skill to a different object (e.g., the lime bucket), the new goal location of the skill will be 5 inches above the current position of the object of interest. Without an object of reference, the DMP goal location remains the final location of the demonstration.

3 Human Subjects Experiment

We conduct an IRB-approved human subjects experiment to investigate the impact of the different modes of feedback provided by the interface on the quality of the demonstrations provided and user experience. The research questions we investigate are:

RQ1) *Does the full system enable participants to be objectively more effective robot teachers, such that participants achieve higher robot*

[1]The implementation we employ is based upon https://pypi.org/project/movement-primitives/0.4.0/.

task completion scores with less teaching time?
RQ2) *Does the full system subjectively bolster alignment between user-predicted robot performance and actual robot task completion?*
RQ3) As an exploratory question, we also ask *How does participant usage of the feedback relate to objective and subjective outcomes, and what individual differences predict feedback usage?*

3.1 Conditions

We conduct a 4x1 between-subjects experiment. The feedback condition, determining which feedback types are provided, is the independent variable:

(1) *Base Interface (Neither):* The interface allows the demonstrator to break down the task into skills, record demonstrations for each skill, and pick skills to assign to each task. The only form of feedback provided to the demonstrator throughout the process is RRR teaching feedback, available in phases 2 and 3, enabling learned skill and learned task replay.
(2) *Augmented Reality (AR):* In addition to the Base Interface, AR teaching feedback is available in phases 2 and 3, including the ability to view a learned skill, skill generalizability, and a learned task (chained skills assigned to a task).
(3) *Foundation Model (FM)*: In addition to the Base Interface, FM feedback is available in phases 1 and 2, including FM planning feedback on the task decomposition and FM teaching feedback on the demonstrated skills.
(4) *Full Interface (Both):* In addition to the Base Interface, all forms of feedback are available.

The ordering of experimental domains (denoted domain ordering) is also randomized and counterbalanced between conditions. All participants experience a practice domain (block touching) first, with no feedback given to participants. The two experimental domains employed in this work – table setting (i.e., set the table for two people where each person has a different preference of utensil placement) and potting soil mixing (i.e., create soil mixtures composed of different quantities of three materials: sand, lime, and manure) – are drawn from prior work [16].

3.2 Policy Evaluation Scenarios

In each domain, we evaluate the learned policy on two known and four held-out scenarios. A detailed description of the six tasks and images of the two environment configurations in each domain can be found in Table 1 and Figure 5 of the Appendix.
Known: These are in-distribution tasks that the participant knows about when teaching, for a set environment configuration.
Task-Generalizing (held-out): After the participant concludes the teaching portion of the study in all domains, participants are asked to use the skills they taught to create recipes (Phase 3) for a set of two novel held-out tasks, which take place in the same environment configuration, but can only be accomplished if the learned skills are properly abstracted.
Environment-Generalizing (held-out): A third set of two tasks consists of the same tasks as the known tasks, in a new environment configuration (i.e., same objects, different locations/orientations). As the task was already presented to the participant in a different environment configuration, the skills assigned to this new environment configuration are auto-populated for the participant to evaluate the robustness of the policy to changing item locations.

3.3 Metrics

We administer a pre-study questionnaire containing the Mini-IPIP Personality survey [6], the Negative Attitudes Towards Robots Scale (NARS) [17], and demographic information (age, gender, racial group, education, field of work/study, and prior computer experience measured by the Computer Usage Checklist [18] and prior robotics experience [16]).

When participants engage with the system we track the number of times each feedback type is used in each domain, and teaching duration for each domain. To evaluate participant perceived robot performance we employ a hand-crafted 4-item Likert Scale with five options ranging from Strongly Disagree (1) to Strongly Agree (5), with an I don't know (0) option. We provide the list of questions and the scale's Cronbach's α in Section B.1.1 of the Appendix.

We administer a post-study questionnaire containing the NASA Task Load Index (NASA-TLX) [9], Multi-Dimensional-Measure of Trust (MDMT) [26], and System Usability Scale [3]. Post-hoc, the experimenters run the assigned recipe for each task on the real robot, annotating the ground truth binary success of discrete checkpoints (Table 2 of the Appendix) to obtain percent task completion.

3.4 Procedure

Participants first completed a practice domain to familiarize themselves with kinesthetic teaching, then taught a Kinova Jaco Gen2 robot two tasks in each domain under their assigned feedback condition. Teaching was limited to 40 minutes per domain. After teaching, participants composed task recipes (Phase 3) for the held-out tasks, and we collected participants' subjective feedback.

4 Results and Discussion

We report results from n=48 (6 participants $\times$ 4 feedback conditions $\times$ 2 domain orderings) (61.7% female; mean age = 26.0; standard deviation = 3.46). Each participant was compensated $40 for completing the two and a half hour study. We report the results of our study and establish statistical significance at the $\alpha = 0.05$ level.

In our analysis we verify the assumption of normality of residuals using the Shapiro-Wilk test and the assumption of homogeneity of variance using Levene's Test. If the data pass these assumptions we employ an Analysis of Variance (ANOVA). When determining subsequent pairwise comparisons, we employ Tukey's HSD. When correlations are evaluated, if the data passes the aforementioned assumptions, we employ Pearson's correlation. If the data fail to pass assumptions of normality of residuals and homogeneity of variance, or if the data is not inherently normally distributed (ordinal or count data) [25], we employ non-parametric equivalents for the aforementioned tests: the Kruskal-Wallis rank sum test with post-hoc Dunn's test for pairwise comparisons, or the Spearman's rank correlation coefficient. Finally, when applicable, we apply Bonferroni Correction to control the family-wise error rate and reduce the chance of making a Type I (false positive) error. In this section,

we summarize the significant findings. Full statistical results, including non-significant comparisons, and participant quotes are provided in Appendix Sections C - D for completeness.

4.1 RQ1

In RQ1 we investigate whether the full system enables participants to be objectively more effective robot teachers such that participants achieve a higher task completion scores with less teaching time.

RQ1.a. We first investigate whether the Both condition outperforms the AR, FM, and Baseline (Neither) conditions across all tasks and for each task type (known tasks and generalization tasks including task generalization and environment generalization) in terms of task completion. We apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{5} = 0.01$. The impact of feedback type on task completion is depicted in Fig. 4.

All Tasks When we aggregated participant task completion across all tasks for each domain assumptions of residual normality (p = 0.116) and homogeneity of variance (p = 0.098) were satisfied. As such we performed a three-way ANOVA on average task completion, with the fixed effects of feedback condition, domain, and domain condition, and the feedback condition x domain interaction term. We find that the main effect of Feedback is significant (F(3,87) = 8.67, p < .001). Tukey's HSD showed that participants in the Both (M=0.556) feedback condition had significantly higher task completion scores than participants in the AR (M = 0.319, p = 0.009, $d = 0.972$), FM (M = 0.313, p = 0.007, $d = 0.997$), and Neither (M = 0.193, p < 0.001, $d = 1.47$) feedback condition.

An ANOVA on task completion with predictors for feedback and task type (with interaction effects) shows that the Both condition outperforms all ablations, and this effect is not dependent upon task type. We find a main effect of feedback (F(3,44)=7.10, p<0.001, generalized eta-squared (ges) =0.276) and task type (F(1.36,59.88)=46.26, p<0.001, ges=0.181) but the feedback-task type interaction effect is not significant (F(4.08,59.88)=0.95, p=0.442, ges=0.013). Tukey's HSD showed that participants in the Both (M=0.556) feedback condition had significantly higher task completion scores than participants in the AR (M=0.319, p =0.026, d=0.693), FM (M=0.312, p=0.021, d=0.711), and Neither (M= 0.193, p<0.001, d=1.06) feedback conditions. Thus, the full system enables participants to be objectively more effective robot teachers in terms of task completion across task types.

Known Tasks As assumptions of residual normality (p = 0.007) and homogeneity of variance (p = 0.535) were not satisfied we employed a Kruskal-Wallis rank sum test on average task completion across feedback conditions. We found a significant effect of feedback on average task completion (χ^2 (3) = 11.5, p = 0.009). Dunn's test showed that participants in the Both feedback condition had significantly higher task completion scores than participants in the Neither feedback condition (Z = 3.32, p = 0.005).

Generalization Tasks As assumptions of residual normality (p = 0.004) and homogeneity of variance (p = 0.023) were not satisfied we employed a Kruskal-Wallis rank sum test on average task completion across feedback conditions. We found a significant effect of feedback on average task completion (χ^2 (3) = 22.5, p < 0.001). Dunn's test showed that participants in the Both feedback condition had significantly higher task completion scores than participants in the Neither feedback condition (Z = 4.71, p < 0.001).

Task Generalization Tasks As assumptions of residual normality (p = 0.022) and homogeneity of variance (p = 0.060) were not satisfied we employed a Kruskal-Wallis rank sum test on average task completion across feedback conditions. We found a significant effect of feedback on average task completion (χ^2 (3) = 19.9, p < 0.001). Dunn's test showed that participants in the Both feedback condition had significantly higher task completion scores than participants in the Neither feedback condition (Z = 4.39, p < 0.001).

Environment Generalization Tasks As assumptions of residual normality (p < 0.001) and homogeneity of variance (p = 0.052) were not satisfied we employed a Kruskal-Wallis rank sum test on average task completion across feedback conditions. We found a significant effect of feedback on average task completion (χ^2 (3) = 25.7, p < 0.001). Dunn's test showed that participants in the Both feedback condition had significantly higher task completion scores than participants in the FM (Z = 3.57, p = 0.002) and Neither feedback condition (Z = 4.89, p < 0.001).

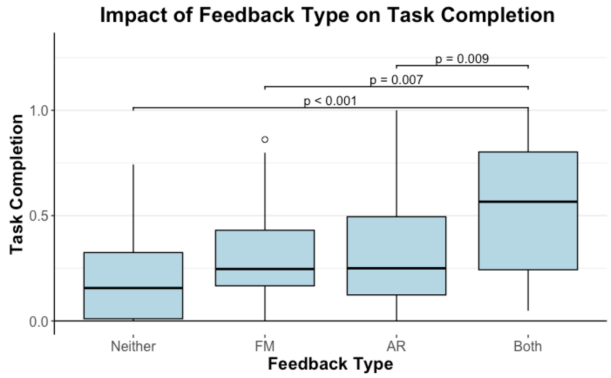


Figure 4: Impact of feedback type on policy performance across all tasks. The Both feedback condition rates significantly higher on task completion compared to the Neither, FM, and AR feedback conditions.

RQ1.a Takeaway: *We find that participants in the Both condition outperform participants in the AR, FM, and Neither feedback conditions across all tasks and that this effect is not dependent upon task type. As expected when subdividing data by task type, we observe lower statistical power. We find that the Both condition outperforms the Neither condition across all task types (known, environment generalization, and task generalization). Additionally, the Both condition outperforms the FM-condition in environment generalization tasks.* We report non-significant comparisons in the Appendix.

RQ1.b. We investigate whether the Both condition will outperform the AR, FM, and Baseline (Neither) conditions in terms of teaching time. Details about teaching duration pre-processing can be found in Section B of the Appendix. As assumptions of residual normality (p < 0.001) and homogeneity of variance (p < 0.001) were not satisfied, we employed a Kruskal-Wallis rank sum test on teaching duration across feedback conditions. We found a significant effect of feedback on teaching duration (χ^2 (3) = 10.1, p = 0.018). Dunn's test showed that participants in the Both feedback condition

had significantly higher teaching duration than participants in the Neither (Z = 2.77, p = 0.033) and AR (Z = 2.74, p = 0.037) feedback condition.

RQ1.b Takeaway: *Participants in the Both condition took longer to teach compared to participants in the Neither and AR conditions.*

RQ1 Takeaway: *The combination of AR and FM feedback improves the performance of taught skills, and the ability of those skills to successfully generalize to task and environment changes. This added performance and generalizability comes at the cost of increased teaching duration. This is an expected tradeoff in systems that support generalization, as users spend more time refining skills based on feedback.*

4.2 RQ2

In RQ2 we investigate whether the full system subjectively bolsters alignment between user-predicted robot performance and actual robot task completion. Details about data pre-processing to obtain misalignment scores can be found in Appendix Section B. As our calculated misalignment is ordinal, we employ Kruskal-Wallis rank sum test to investigate this research question. As we investigate the impact of feedback condition on alignment for all tasks, known tasks, and generalizability tasks, we apply Bonferroni Correction and report results with respect to a significance level of $\frac{0.05}{3} = 0.017$.

Alignment on All Tasks We employ a Kruskal-Wallis rank sum test on participants' overall misalignment score across feedback conditions. We found a significant effect of Feedback on participants' misalignment (χ^2 (3) = 10.9, p = 0.012). Dunn's test showed that participants in the Both condition had significantly higher alignment than those in the Neither condition (Z = 3.11, p = 0.011).

Alignment on Known Tasks We employ a Kruskal-Wallis rank sum test on an participants' overall misalignment score across feedback conditions and do not find a significant effect of feedback condition on alignment.

Alignment on Generalization Tasks We employ a Kruskal-Wallis rank sum test on an participants's overall misalignment score across feedback conditions. We found a significant effect of Feedback on misalignment (χ^2 (3) = 10.7, p = 0.013). Dunn's test showed that participants in the Both condition had significantly higher alignment than those in the Neither condition (Z = 3.25, p = 0.007).

RQ2 Takeaway: *The combination of AR and FM feedback improves alignment between true robot performance and predicted robot performance across all tasks and across generalization tasks compared to just RRR.*

4.3 RQ3

In RQ3 we conduct an exploratory investigation of how participant usage of the feedback tools relate to objective and subjective outcomes, and what individual differences predict usage.

RQ3.a Objective Outcomes. We first investigate whether the degree of participant usage of the available feedback tools objectively positively correlate with task completion scores. As feedback usage is count data, we employ Spearman's rank correlation coefficient on each of these three feedback sources. We apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{5} = 0.01$. We found a significant positive correlation between AR feedback usage and task completion (ρ = 0.212, n = 432, p < 0.001), suggesting that increased AR usage was associated with higher task completion scores. We further found a significant positive correlation between FM planning feedback usage and task completion (ρ = 0.134, n = 432, p = 0.005), suggesting that increased FM planning feedback usage was associated with higher task completion scores. Because FM Teaching feedback was not correlated with task completion, it is treated as an ablation rather than part of the final system. We report this result for completeness, and our final system consists of FM Planning and AR Teaching feedback only.

RQ3.a Takeaway: *Increased usage of AR teaching and FM planning feedback are correlated with higher robot performance. However, use of FM teaching feedback had no correlation with robot performance.*

RQ3.b Subjective Outcomes. We next investigate whether the degree of participant usage of the available feedback tools subjectively impact participant perceptions of and attitudes toward the system, including workload, learned trust, and usability. As we investigate the correlation between each of these three factors and the three forms of feedback, we apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{9} = 0.0056$.

Workload: We employ Spearmans's rank correlation coefficient on each feedback usage condition and observe a significant positive correlation between RRR feedback usage and workload (ρ = 0.192, n = 432, p < 0.001), suggesting that increased RRR usage was associated with higher workload.

Learned Trust: We employ Spearmans's rank correlation coefficient on each feedback usage condition. We observe a significant positive correlation between AR feedback usage and learned trust (ρ = 0.140, n = 432, p = 0.004), suggesting that increased AR usage was associated with higher learned trust. We observe a significant positive correlation between FM feedback usage and learned trust (ρ = 0.146, n = 432, p = 0.002), suggesting that increased FM usage was associated with higher learned trust. We observe a significant negative correlation between RRR feedback usage and learned trust (ρ = -0.177, n = 432, p < 0.001), suggesting that increased RRR usage was associated with lower learned trust.

Usability: We employ Spearman's rank correlation coefficient on each feedback usage condition, but do not find a significant correlation between AR, FM, or RRR feedback usage and usability.

RQ3.b Takeaway: *Increased use of AR and FM feedback increases learned trust. Increased use of RRR feedback decreases learned trust and increases workload.*

RQ3.c Predictors of Usage. We conduct an exploratory investigation of predictors of participants' degree of usage of the feedback tools provided, including personality traits, negative attitudes towards robots, and prior experience. Identifying such participant traits can (1) inform future human-robot interaction design practices, and (2) ensure that inferences about the use of such systems are not confounded by unmodeled, easily measured sources of variance. As feedback usage is count data, to address this research question we employ Spearman's rank correlation coefficient on

each of these three feedback sources, filtering the data such that each correlation only includes participants with access to the form of feedback.

Personality: We employ Spearman's rank correlation coefficient on the usage of each feedback type for each of the five personality subscales and apply Bonferroni Correction, reporting results with respect to a significance level of $\frac{0.05}{5\times3} = 0.003$. We find a significant positive correlation between RRR feedback usage and agreeableness (ρ = 0.170, n = 432, p < 0.001) as well as openness (ρ = 0.158, n = 432, p = 0.001) and a significant negative correlation between RRR feedback usage and conscientiousness (ρ = -0.235, n = 432, p < 0.001). **Takeaway:** *High conscientiousness scores are correlated with decreased RRR feedback usage. High agreeableness and openness scores are associated with increased RRR feedback usage.*

Negative Attitudes: We employ Spearman's rank correlation coefficient on the usage of each feedback type for each of the three NARS subscales (Negative Situations of Interaction with Robots (HRI), Negative Social Influence, and Negative Emotions in HRI) and the NARS scale overall and apply Bonferroni Correction, reporting results with respect to a significance level of $\frac{0.05}{4\times3} = 0.004$. We found a significant negative correlation between AR feedback usage and the NARS Scale (ρ = -0.263, n = 288, p < 0.001), the Negative Attitudes toward Situations of HRI subscale (ρ = -0.233, n = 288, p < 0.001), and the Negative Attitudes toward Emotions in HRI subscale (ρ = -0.231, n = 288, p < 0.001). We found a significant negative correlation between RRR feedback usage and the Negative Attitudes toward Situations of HRI subscale (ρ = -0.160, n = 432, p < 0.001) as well as the Negative Attitudes toward the Social Influence of Robots subscale (ρ = -0.255, n = 432, p < 0.001). However we found a significant positive correlation between RRR feedback usage and the Negative Attitudes toward Emotions in HRI subscale (ρ = 0.155, n = 432, p = 0.001). **Takeaway:** *Less negative attitudes toward the robot are correlated with higher AR usage. The impact of negative attitudes on RRR usage is more nuanced, with two subscales negatively and one subscale positively correlated with RRR usage.*

Prior Experience: We employ Spearman's rank correlation coefficient on the usage of each feedback type for the prior robotics experience scale and the prior computer experience scale and apply Bonferroni Correction, reporting results with respect to a significance level of $\frac{0.05}{2\times3} = 0.0083$. We found a significant positive correlation between prior robotics experience and AR feedback usage (ρ = 0.227, n = 288, p < 0.001), suggesting that less prior robotics experience is associated with decreased AR feedback usage. **Takeaway:** *Prior robotics experience is positively correlated with AR feedback usage.*

RQ3.c Takeaway: *Negative attitudes towards robots influence the use of RRR and AR feedback. Personality traits influence the use of RRR feedback. Additionally, prior robotics experience positively influences AR feedback use.*

RQ3 Takeaway: *Participant usage of the final system (composed of FM planning and AR teaching feedback) is correlated with objective outcomes (robot performance). Participant usage of AR and FM feedback is correlated with subjective outcomes (learned trust). Negative attitudes towards robots, personality traits, and prior robotics experience influence feedback usage.*

4.4 Limitations and Future Work

When visualizing a full task in AR, all skills are learned at once and then deployed to the headset. Because of this, if any skills alter the environment, these changes are not registered or visualized for subsequent skills. For our user study, we assume that any object grasped shares the location of the end effector until it is dropped (thus assuming that the grasp was successful). This ensures that the location of the item is known for later skills, whose goal location may be defined relative to that item. Future work could consider employing a simulator to relax the assumption of a successful grasp.

This work employs AprilTags for object detection, and includes object labels to facilitate FM scene understanding. Future work could develop a more robust sensing and perception pipeline by incorporating object classifiers and semantic labeling/segmentation to input explicit scene observations into the Image Descriptor VLM prompt to reduce incorrect item identification and VLM hallucination rate. This would likely improve the utility of the FM teaching feedback, as one of our system failure cases was instances where the FM teaching feedback pipeline would hallucinate or incorrectly identify objects in the environment. Future work could additionally explore structured ways for non-expert end users to provide custom initial environment and domain descriptions to LLM or VLM prompts, thus improving the flexibility of FM feedback.

5 Conclusion

We develop an interactive feedback system that provides bespoke, autonomously generated feedback to non-expert demonstrators, enabling higher objective policy performance compared to system baselines in a learning from demonstration (LfD) paradigm. Our novel LfD system provides guidance both to inform the process of planning what skills to teach, and the process of demonstrating those skills. We generate this guidance with foundation models, which help evaluate the necessity and sufficiency of skills, and through augmented reality visualization, which helps evaluate skill generalizability and enables lightweight debugging. We empirically validate our system's benefits compared to ablated baseline systems: AR feedback alone, FM feedback alone, and a baseline system with neither AR nor FM feedback but where learned skills can be played physically on the robot. We observe positive results for our final system that synthesizes FM planning and AR teaching feedback on human-provided demonstration quality (in terms of measured robot performance on seen and unseen tasks), learned trust, and alignment between measured robot performance and user-predicted robot performance.

Acknowledgments

This research was supported by NSF grants IIS-2340177 and IIS-2112633 as well as a gift from Konica Minolta.

References

[1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the Opportunities and Risks of Foundation Models. *ArXiv* (2021). https://crfm.stanford.edu/assets/report.pdf

[3] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[4] Maya Cakmak and Leila Takayama. 2014. Teaching people how to teach robots: The effect of instructional materials and dialog design. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. 431–438.

[5] Sonia Chernova and Andrea L Thomaz. 2022. *Robot learning from human teachers*. Springer Nature.

[6] M Brent Donnellan, Frederick L Oswald, Brendan M Baird, and Richard E Lucas. 2006. The mini-IPIP scales: tiny-yet-effective measures of the Big Five factors of personality. *Psychological assessment* 18, 2 (2006), 192.

[7] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. 2023. Vision-Language Models as Success Detectors. In *Proceedings of The 2nd Conference on Lifelong Learning Agents (Proceedings of Machine Learning Research, Vol. 232)*, Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup (Eds.). PMLR, 120–136. https://proceedings.mlr.press/v232/du23b.html

[8] Nakul Gopalan, Nina Moorman, Manisha Natarajan, and Matthew C Gombolay. 2022. Negative Result for Learning from Demonstration: Challenges for End-Users Teaching Robots with Task And Motion Planning Abstractions.. In *Robotics: Science and Systems*.

[9] S Hart. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload/Elsevier* (1988).

[10] Bryce Ikeda, Maitrey Gramopadhye, LillyAnn Nekervis, and Daniel Szafir. 2025. Marcer: Multimodal augmented reality for composing and executing robot tasks. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 529–539.

[11] Bryce Ikeda and Daniel Szafir. 2022. Advancing the design of visual debugging tools for roboticists. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 195–204.

[12] Michael Kaiser, Holger Friedrich, and Rudiger Dillmann. 1995. Obtaining good performance from a bad teacher. In *Programming by Demonstration vs. Learning from Examples Workshop at ML*, Vol. 95. Citeseer.

[13] Matthew B Luebbers, Connor Brooks, Carl L Mueller, Daniel Szafir, and Bradley Hayes. 2021. Arc-lfd: Using augmented reality for interactive long-term robot skill maintenance via constrained learning from demonstration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3794–3800.

[14] Fiona Luo. 2024. Vision-language models for robot success detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 23750–23752.

[15] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* 36 (2023), 46534–46594.

[16] Nina Marie Moorman, Nakul Gopalan, Aman Singh, Erin Hedlund-Botti, Mariah L Schrum, Chuxuan Yang, Lakshmi Seelam, and Matthew Gombolay. 2023. Investigating the impact of experience on a user's ability to perform hierarchical abstraction. In *Robotics: Science and Systems*.

[17] Tatsuya Nomura, Tomohiro Suzuki, Takayuki Kanda, and Kensuke Kato. 2006. Measurement of negative attitudes toward robots. *Interaction Studies. Social Behaviour and Communication in Biological and Artificial Systems* 7, 3 (2006), 437–454.

[18] Annalyse Callahan Raub. 1981. *Correlates of computer anxiety in college students*. University of Pennsylvania.

[19] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. 2020. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems* 3, 1 (2020), 297–330.

[20] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. 2020. Communicating robot arm motion intent through mixed reality head-mounted displays. In *Robotics research: The 18th international symposium ISRR*. Springer, 301–316.

[21] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2998–3009.

[22] Daeun Song, Jing Liang, Xuesu Xiao, and Dinesh Manocha. 2025. Vl-tgs: Trajectory generation and selection using vision language models in mapless outdoor environments. *IEEE Robotics and Automation Letters* (2025).

[23] Halit Bener Suay, Russell Toris, and Sonia Chernova. 2012. A practical comparison of three robot learning from demonstration algorithm. *International Journal of Social Robotics* 4 (2012), 319–330.

[24] Yiliu Tang, Jason Situ, Andrea Yaoyun Cui, Mengke Wu, and Yun Huang. 2025. LLM Integration in Extended Reality: A Comprehensive Review of Current Trends, Challenges, and Future Perspectives. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–24.

[25] Miranda A Too, Udi Alter, and David B Flora. 2025. Modelling Count Data in Psychological Research: An Applied Tutorial. *International Journal of Psychology* 60, 2 (2025), e70018.

[26] Daniel Ullman and Bertram F Malle. 2019. Measuring gains and losses in human-robot trust: Evidence for differentiable components of trust. In *2019 14th ACM/IEEE international Conference on human-robot interaction (HRI)*. IEEE, 618–619.

[27] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. 2018. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 316–324.

[28] Carl Winge, Adam Imdieke, Bahaa Aldeeb, Dongyeop Kang, and Karthik Desingh. 2024. Talk through it: End user directed manipulation learning. *IEEE Robotics and Automation Letters* (2024).

Received 2025-09-30; accepted 2025-12-23

1 Tasks

In this section we list the domains and tasks of this study (Table 1) and the scoring rules for obtaining task completion (Table 2). The two experimental domains and their respective environment configurations are depicted in Figure 1.

Table 1: Task and environment configurations (designed with varying object locations and orientations).

	Environment	Block Touching	Soil Mixing	Table Setting
Known	Config 1	Touch the red block, then touch the green block.	Make a soil mixture with one scoop of sand and one scoop of lime.	Set a fork to the right of the plate for Person 1.
	Config 1	Touch the green block, then touch the blue block.	Make a soil mixture with two scoops of lime and one scoop of sand.	Set a fork to the right of the plate for Person 1, and a fork to the left of the plate for Person 2.
Task Generalizing	Config 1	N/A	Make a soil mixture with one scoop of sand, one scoop of lime, and one scoop of manure.	Set a fork to the right of the plate for Person 1, and set a fork to the left and a knife to the right of the plate for Person 2.
	Config 1	N/A	Make a soil mixture with two scoops of sand and one scoop of manure.	Set a fork to the left and a knife to the right of the plate for Person 1, and set a fork to the right of the plate for Person 2.
Env. Generalizing	Config 2	N/A	Make a soil mixture with one scoop of sand and one scoop of lime.	Set a fork to the right of the plate for Person 1.
	Config 2	N/A	Make a soil mixture with two scoops of lime and one scoop of sand.	Set a fork to the right of the plate for Person 1, and a fork to the left of the plate for Person 2.

2 Data Pre-Processing and Statistical Analysis

2.1 Predicted Robot Performance Likert Scale

Our hand-crafted scale questions include:

(1) The robot will succeed at accomplishing the task with the recipe I created.
(2) The robot's execution of this task will be as I intended.
(3) The robot understood what I wanted it to do for this task.
(4) The robot will not perform the task well, given my demonstrations. (-)

2.1.1 Cronbach's Alpha. As the reliability of a scale may be time-dependent we report the Cronbach's α for the Predicted Robot Performance Likert scale for each task and each domain, depicted in Table 3. We find that all α values are above 0.6 (ranging from 0.643 to 0.827).

Table 2: Task Breakdown and Task Completion Scoring

Domain	Example Task	Optimal Breakdown	Max Task Completion Score
Soil Mixing	Create a plant mixture with one scoop of sand and two scoops of manure.	(1) Go to [sand] (2) Scoop [sand] (3) Carry to [mixing bucket] (4) Drop in [mixing bucket] (5) Go to [manure] (6) Scoop [manure] (7) Carry to [mixing bucket] (8) Drop in [mixing bucket] (9) Go to [manure] (10) Scoop [manure] (11) Carry to [mixing bucket] (12) Drop in [mixing bucket]	• +1 if scoop moves above material bucket, within 6 inches of the top of the bucket • +1 if scoop enters the space of the material bucket, with a motion to gather material • +1 if scoop does not tilt and moves above mixing bucket • +1 if scoop tilts into mixing bucket
Table Setting	Set a fork to the right of the plate for Person 1 and a fork to the left of the plate for Person 2.	(1) Go to [fork 1] (2) Pick up [fork 1] (3) Go to right [plate 1] (4) Drop utensil [fork 1] (5) Go to [fork 2] (6) Pick up [fork 2] (7) Go to left [plate 2] (8) Drop utensil [fork 2]	• +1 if end effector moves within 6 inches of utensil with open gripper • +1 if end effector closes while touching utensil • +1 if end effector moves over correct utensil target location (left or right of plate) • +1 if end effector opens over correct utensil target location

Table 3: Cronbach's α for Perceived Performance Likert Scale

Domain	Task	Cronbach's α
Soil	T1	0.816
Soil	T2	0.803
Soil	T3	0.658
Soil	T4	0.681
Soil	T5	0.787
Soil	T6	0.812
Table	T1	0.775
Table	T2	0.757
Table	T3	0.688
Table	T4	0.643
Table	T5	0.753
Table	T6	0.827

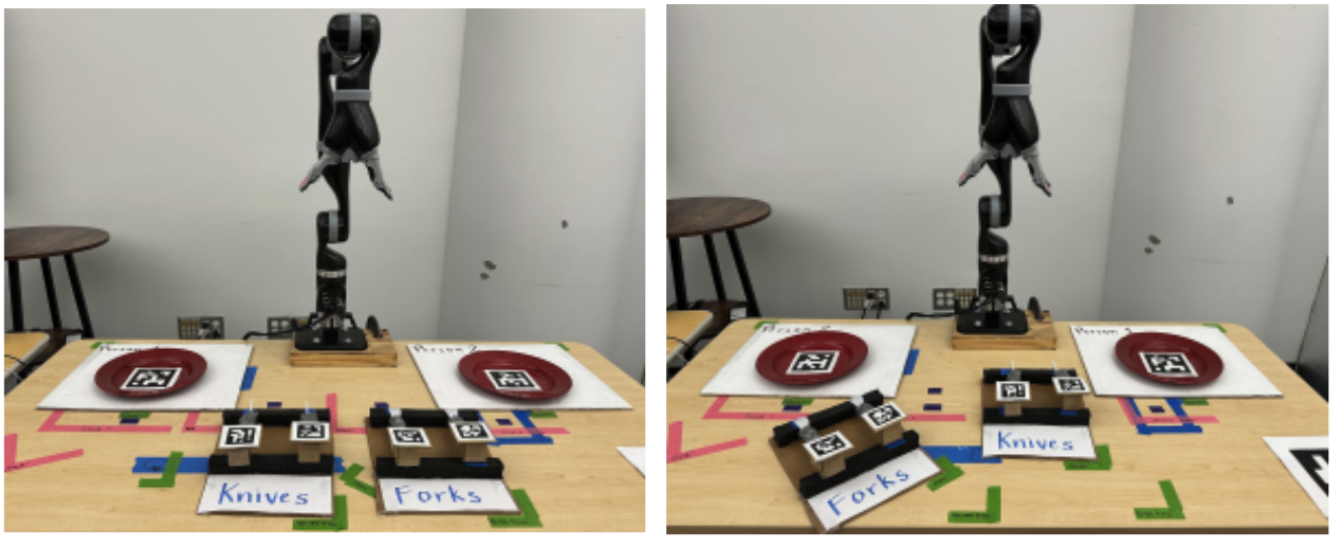
2.2 Misalignment

To calculate misalignment between user-predicted performance and true policy performance (task completion) we obtain participants' prediction for a task by summing their Predicted Robot Performance Likert scale responses together, and then normalizing by the maximum possible score, to get a value between 0 and 1. Then, we take the absolute difference with the actual robot task completion score from that task (also between 0 and 1).

2.3 Teaching Duration

Teaching duration is automatically measured by the interface through obtaining timestamps at the beginning and end of teaching for a domain. However, the 40 minute time limit for teaching duration was enforced by an analog timer in the experiment space, which was paused temporarily on rare occasions to resolve technical difficulties during the study (e.g., restarting the robot after an e-stop). As this time spent paused was not the result of added teaching time, the teaching duration for each domain is capped at 40 minutes, even if the

(a) Soil mixing with configuration 1 on the left and configuration 2 on the right.



(b) Table setting with configuration 1 on the left and configuration 2 on the right.

Figure 1: Environment configurations for both domains

teaching interface was active for longer than 40 minutes. Additionally, the collected teaching duration data for the soil mixture domain for one participant was corrupted, and is not included in this analysis.

3 Complete Results

Tables 5 - 10 contain both positive and negative results from our statistical analysis of RQ1 - RQ3, and Table 4 reports the mean task completion for all feedback conditions when aggregating across different task types.

3.1 Bonferroni Correction

When running multiple hypothesis tests on shared data, we apply Bonferroni Correction to control the family-wise error rate and reduce the chance of making a Type I (false positive) error.

Table 4: Mean task completion (i.e., the fraction of discrete task checkpoints successfully accomplished by the robot, between [0,1], using participant-assigned skills and recipes) by feedback condition across different task groupings.

Feedback Condition	All Tasks	Known Tasks	Task Generalization Tasks	Environment Generalization Tasks
Both	0.556	0.661	0.571	0.435
AR	0.319	0.428	0.278	0.251
FM	0.313	0.462	0.319	0.156
Neither	0.193	0.333	0.184	0.063

3.1.1 RQ1. In RQ1.a we investigate whether the Both condition outperforms the AR, FM, and Baseline (Neither) conditions across all tasks and for each task type (known tasks and generalization tasks, including task generalization and environment generalization) in terms of task completion. We apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{5} = 0.01$.

3.1.2 RQ2. In RQ2, we investigate whether the full system subjectively bolsters alignment between user-predicted robot performance and actual robot task completion for all tasks, known tasks, and generalization tasks. We apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{3} = 0.017$.

3.1.3 RQ3. In RQ3.a, we investigate whether the degree of participant usage of the available feedback tools (RRR, AR, and FM with subsequent analysis for both FM teaching, FM planning) objectively positively correlate with task completion scores. We apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{5} = 0.01$.

In RQ3.b, we investigate whether the degree of participant usage of the available feedback tools (RRR, AR, and FM) subjectively impact participant perceptions of and attitudes toward the system, including workload, learned trust, and usability. We apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{3\times3} = 0.0056$.

In RQ3.c, we conduct an exploratory investigation of predictors of participants' degree of usage of the feedback tools provided (RRR, AR, and FM), including personality traits, negative attitudes towards robots, and prior experience. For the five personality traits, we apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{5\times3} = 0.003$. For negative attitudes towards robots, we investigate the overall scale as well as its three subscales and apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{4\times3} = 0.004$. Finally, for prior robotics and computer science experience we apply Bonferroni Correction, and report results with respect to a significance level of $\frac{0.05}{2\times3} = 0.0083$.

Table 5: This table details the dependent and independent variables, covariates, statistical tests, test assumptions, test statistics, and p-values for RQ1.a (all tasks). Bonferroni corrections are described in the Section 3.1.

RQ1.a: Impact of Feedback Condition on Task Completion							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Task Completion	Feedback × Domain + Domain Condition	All Tasks	Shapiro-Wilk	N/A	N/A	N/A	$p = .116$
			Levene				$p = .098$
			ANOVA	Feedback	N/A	$F(3, 87) = 8.67$	$p < .001$
				Domain		$F(1, 87) = .552$	$p = .459$
				Domain Condition		$F(1, 87) = 3.94$	$p = .050$
				Feedback: Domain		$F(3, 87) = .709$	$p = .549$
			TukeyHSD	N/A	Neither–FM	$d = -.472$	$p = .365$
					Neither–AR	$d = -.497$	$p = .319$
					Neither–Both	$d = -1.47$	$p < .001$
					FM-AR	$d = -.025$	$p = 1.00$
					FM–Both	$d = -.997$	$p = .007$
					AR–Both	$d = -.972$	$p = .009$
Task Completion	Feedback × Task Type + Error(ID/ Task Type)	All Tasks (aggregated by task type)	ANOVA	Feedback	N/A	$F(3, 44) = 7.10$	$p < .001$
				Task Type		$F(1.36, 59.9) = 46.3$	$p < .001$
				Feedback: Task Type		$F(4.08, 59.9) = .950$	$p = .442$
			TukeyHSD	N/A	AR–Both	$d = -.693$	$p = .026$
					AR–FM	$d = .019$	$p = 1.00$
					AR–Neither	$d = .367$	$p = .413$
					Both–FM	$d = .711$	$p = .021$
					Both–Neither	$d = 1.06$	$p < .001$
					FM–Neither	$d = .349$	$p = .459$

Table 6: This table details the dependent and independent variables, covariates, statistical tests, test assumptions, test statistics, and p-values for RQ1.a sub analysis on task types. Bonferroni corrections are described in Section 3.1.

RQ1.a: Impact of Feedback Condition on Task Completion							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Task Completion	Feedback	Known	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 11.5$	$p = .009$
			Dunn	N/A	AR–Both	$Z = -2.26$	$p = .143$
					AR–FM	$Z = -.426$	$p = 1.00$
					Both–FM	$Z = 1.83$	$p = .401$
					AR–Neither	$Z = 1.06$	$p = 1.00$
					Both–Neither	$Z = 3.32$	$p = .005$
					FM–Neither	$Z = 1.49$	$p = .818$
Task Completion	Feedback	Generalization	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 22.5$	$p < .001$
			Dunn	N/A	AR–Both	$Z = -2.68$	$p = .044$
					AR–FM	$Z = .169$	$p = 1.00$
					Both–FM	$Z = 2.85$	$p = .026$
					AR–Neither	$Z = 2.03$	$p = .257$
					Both–Neither	$Z = 4.71$	$p < .001$
					FM–Neither	$Z = 1.86$	$p = .380$
Task Completion	Feedback	Task Generalization	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 19.9$	$p < .001$
			Dunn	N/A	AR–Both	$Z = -2.91$	$p = .022$
					AR–FM	$Z = -.483$	$p = 1.00$
					Both–FM	$Z = 2.43$	$p = .091$
					AR–Neither	$Z = 1.47$	$p = .845$
					Both–Neither	$Z = 4.39$	$p < .001$
					FM–Neither	$Z = 1.96$	$p = .303$
Task Completion	Feedback	Env. Generalization	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 25.7$	$p < .001$
			Dunn	N/A	AR–Both	$Z = -2.51$	$p = .072$
					AR–FM	$Z = 1.05$	$p = 1.00$
					Both–FM	$Z = 3.57$	$p = .002$
					AR–Neither	$Z = 2.38$	$p = .105$
					Both–Neither	$Z = 4.89$	$p < .001$
					FM–Neither	$Z = 1.33$	$p = 1.00$

Table 7: This table details the dependent and independent variables, covariates, statistical tests, test assumptions, test statistics, and p-values for RQ1.b and RQ2. Bonferroni corrections are described in Section 3.1.

RQ1.b: Impact of Feedback Condition on Teaching Time							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Teaching Time	Feedback	All Tasks	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 10.1$	$p = .018$
			Dunn	N/A	AR–Both	$Z = -2.74$	$p = .037$
					AR–FM	$Z = -.865$	$p = 1.00$
					Both–FM	$Z = 1.84$	$p = .392$
					AR–Neither	$Z = .037$	$p = 1.00$
					Both–Neither	$Z = 2.77$	$p = .033$
					FM–Neither	$Z = .902$	$p = 1.00$
RQ2: Impact of Feedback Condition on Alignment							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Misalignment	Feedback	All Tasks	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 10.9$	$p = .012$
			Dunn	N/A	AR–Both	$Z = 2.49$	$p = .076$
					AR–FM	$Z = .576$	$p = 1.00$
					Both–FM	$Z = -1.92$	$p = .331$
					AR–Neither	$Z = -.620$	$p = 1.00$
					Both–Neither	$Z = -3.11$	$p = .011$
					FM–Neither	$Z = -1.20$	$p = 1.00$
Misalignment	Feedback	Known	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 8.09$	$p = .044$
			Dunn	N/A	AR–Both	$Z = 2.43$	$p = .091$
					AR–FM	$Z = .525$	$p = 1.00$
					Both–FM	$Z = -1.90$	$p = .342$
					AR–Neither	$Z = -.036$	$p = 1.00$
					Both–Neither	$Z = -2.46$	$p = .082$
					FM–Neither	$Z = -.561$	$p = 1.00$
Misalignment	Feedback	Generalization	Kruskal-Wallis	Feedback	N/A	$\chi^2(3) = 10.7$	$p = .013$
			Dunn	N/A	AR–Both	$Z = 1.81$	$p = .424$
					AR–FM	$Z = -.131$	$p = 1.00$
					Both–FM	$Z = -1.94$	$p = .315$
					AR–Neither	$Z = -1.44$	$p = .893$
					Both–Neither	$Z = -3.25$	$p = .007$
					FM–Neither	$Z = -1.31$	$p = 1.00$

Table 8: This table details the dependent and independent variables, covariates, statistical tests, test assumptions, test statistics, and p-values for RQ3.a and RQ3.b. Bonferroni corrections are described in Section 3.1.

RQ3.a: Impact of Feedback Usage on Objective Outcomes							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Task Completion	AR Feedback Usage	All Tasks	Spearman Correlation	N/A	N/A	$\rho = .212$ $n = 432$	$p < .001$
	FM Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .090$ $n = 432$	$p = .061$
	FM Planning Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .134$ $n = 432$	$p = .005$
	FM Teaching Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = -.023$ $n = 432$	$p = .638$
	RRR Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .027$ $n = 432$	$p = .574$
RQ3.b: Impact of Feedback Usage on Subjective Outcomes							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Workload	AR Feedback Usage	All Tasks	Spearman Correlation	N/A	N/A	$\rho = -.031$ $n = 432$	$p = .527$
	FM Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .064$ $n = 432$	$p = .183$
	RRR Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .192$ $n = 432$	$p < .001$
Trust	AR Feedback Usage	All Tasks	Spearman Correlation	N/A	N/A	$\rho = .140$ $n = 432$	$p = .004$
	FM Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .146$ $n = 432$	$p = .002$
	RRR Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = -.177$ $n = 432$	$p < .001$
Usability	AR Feedback Usage	All Tasks	Spearman Correlation	N/A	N/A	$\rho = -.096$ $n = 432$	$p = .047$
	FM Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .034$ $n = 432$	$p = .485$
	RRR Feedback Usage		Spearman Correlation	N/A	N/A	$\rho = .038$ $n = 432$	$p = .432$

Table 9: This table details the dependent and independent variables, covariates, statistical tests, test assumptions, test statistics, and p-values for RQ3.c's personality analysis. Bonferroni corrections are described in Section 3.1. All analysis is conducted on all tasks; the data scope column indicates which participants' data are used for the test.

RQ3.c: Predictors of Feedback Usage							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Personality							
Extraversion	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = .021$ $n = 288$	$p = .725$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = -.021$ $n = 288$	$p = .728$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = -.058$ $n = 432$	$p = .228$
Agreeableness	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = .052$ $n = 288$	$p = .384$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = -.138$ $n = 288$	$p = .019$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = .170$ $n = 432$	$p < .001$
Conscientiousness	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = .167$ $n = 288$	$p = .005$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = .018$ $n = 288$	$p = .767$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = -.235$ $n = 432$	$p < .001$
Openness	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = -.018$ $n = 288$	$p = .756$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = -.011$ $n = 288$	$p = .852$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = .158$ $n = 432$	$p = .001$
Emotional Stability	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = -.031$ $n = 288$	$p = .601$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = .128$ $n = 288$	$p = .029$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = -.129$ $n = 432$	$p = .007$

Table 10: This table details the dependent and independent variables, covariates, statistical tests, test assumptions, test statistics, and p-values for RQ3.c's NARS and prior experience analysis. Bonferroni corrections are described in Section 3.1. All analysis is conducted on all tasks; the data scope column indicates which participants' data are used for the test.

RQ3.c: Predictors of Feedback Usage							
DV	**IV/Covariates**	**Data Scope**	**Statistical Test**	**Effect**	**Contrasts**	**Test Statistic / Effect Size**	**Test p-value**
Negative Attitudes towards Robotics (NARS) Scale							
NARS	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = -.263$ $n = 288$	$p < .001$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = .135$ $n = 288$	$p = .022$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = -.103$ $n = 432$	$p = .032$
Negative Situations of HRI Subscale	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = -.233$ $n = 288$	$p < .001$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = .063$ $n = 288$	$p = .285$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = -.160$ $n = 432$	$p < .001$
Negative Social Influence Subscale	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = -.158$ $n = 288$	$p = .007$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = .138$ $n = 288$	$p = .019$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = -.255$ $n = 432$	$p < .001$
Negative Emotions in HRI Subscale	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = -.231$ $n = 288$	$p < .001$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = .129$ $n = 288$	$p = .029$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = .155$ $n = 432$	$p = .001$
Prior Experience							
Robotics Experience	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = .227$ $n = 288$	$p < .001$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = -.024$ $n = 288$	$p = .681$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = .112$ $n = 432$	$p = .020$
Computer Science Experience	AR Feedback Usage	Both, AR	Spearman Correlation	N/A	N/A	$\rho = .061$ $n = 288$	$p = .302$
	FM Feedback Usage	Both, FM	Spearman Correlation	N/A	N/A	$\rho = -.076$ $n = 288$	$p = .197$
	RRR Feedback Usage	Both, FM, AR	Spearman Correlation	N/A	N/A	$\rho = .117$ $n = 432$	$p = .015$

4 Participant Quotes

In this section we provide representative quotes from each feedback condition in our study.

4.1 Both

In the Both condition participants had access to FM, AR, and RRR feedback.

Participant 4431 (Both): "Once I learned the system limitations I could work around it... All three [forms of feedback] were useful, the LLM was less useful to me because it told me to split up actions but I didn't trust it cause it didn't say why. AR would prove to me why and it was fast which was important for iteration. Real [RRR feedback] was also important but too slow to rapidly iterate in the limited amount of time."

Participant 1816 (Both): "I feel like the LLM didn't help me much, and this could possibly be from like a person bias that I don't like using LLMs day to day... I do think getting a visual [AR feedback] did help a lot more than explaining a problem that I did wrong [FM feedback]... I liked seeing the robot do it you know in real life [RRR feedback] but it's not necessary unless for AR I'm pretty sure it's working... there's really no need to test it out in real life unless it looks pretty good in augmented and I just want to do a final run to make sure everything works."

4.2 FM

In the FM condition participants had access to FM and RRR feedback.

Participant 6713 (FM): "I was surprised at how hard it was to make the robot work the way that I wanted it to work... the most confusing part for me was understanding how the robot thinks... I don't really have a full mental model about how the robot is thinking about the task still... I felt confused a lot during the study. I think the soil test especially, when the robot was doing things that were completely unexpected was really confusing ... I felt like I kind of got stuck cause it wasn't doing anything like what I was expecting it to do and I couldn't trace what it was actually doing back to like anything that I was doing"

Participant 7067 (FM): "I think it gave very good indicators [feedback] and the evaluation was very helpful in understanding if it understood what I had tasked it to do... I liked the evaluation [RRR] more [than FM] because it was more like, I don't know, it's a physical activity so it's like it helps to have physical feedback."

4.3 AR

In the AR condition participants had access to AR and RRR feedback.

Participant 3779 (AR): "I think the feedback helped a lot when I was training it... I already thought it was going to work ... but turns out that the simulation wasn't as I expected so based on the feedback I have to change how I train the robot."

Participant 3715 (AR): "Without feedback I don't know what the robot actually learned... Seeing the robot actually do [RRR] was the best feedback... even if the AR is a centimeter off it wasn't going to do it right."

4.4 Neither

In the Neither condition participants only had access to RRR feedback.

Participant 3035 (Neither): "I'm not getting it, why it's not doing what I said... Would be nice if I had feedback for each module [task] so I know what I did wrong."

Participant 2310 (Neither): "I would have liked to know more of the *why* certain things were happening, mostly just cause I'm curious, but um yeah I think it would have been helpful to have some sort of like explanation as to what was causing things to not work so that make it easier to figure out."

5 Foundation Model Prompt Content

We used the Chain-of-Thought (CoT) prompting method to ensure the stability of the FM feedback, where the FM is instructed to provide a CoT reasoning following specific guidelines prior to selecting an error class and providing human participants with an explanation.

We used html formatting to separate different stages of the FM response and automate the process. We set the decoding temperature to 0 to minimize randomness and obtain consistent outputs.

5.1 Error Class Descriptions

5.1.1 FM Planning Feedback Error Classes. In addition to "Error Not Listed" and "No Error" classes, the FM planning feedback error classes include:

(1) *Missing Skill*: This error class represents a skill (or set of skills) that is needed to accomplish the task (given the existing set of skills) that is missing.
(2) *Irrelevant Skill*: Informed by the Incorrect Action introduced in Kaiser et al. [2], this error class represents skills that do not contribute to achieving tasks in this domain.
(3) *Redundant Skill*: Informed by the redundancy metric [3] and Unnecessary Action source of degradation introduced in Kaiser et al. [2], this error class represents skills that do not contribute novel functionality towards achieving the tasks in this domain, given the existing skills.
(4) *Overly-General Skill*: Informed by the Abstraction score metric defined in prior work [1] as well as the Wrongly Specified Intention (incomplete necessary conditions set) source of degradation introduced in Kaiser et al. [2], this error class represents skills that are so abstracted as to not be useful towards accomplishing the task (e.g. move up).
(5) *Overly-Specific Skill*: Informed by the observed under-abstraction source of novice demonstrator suboptimality in our prior work [3] as well as the Wrongly Specified Intention (insufficient specification resulting in overly complex post-condition) source of degradation introduced in Kaiser et al. [2], this error class represents skills that are not general enough to apply to slight variations of the task.

The way in which we describe each of the FM planning feedback error classes in the prompt can be found in Figure 2.

5.1.2 FM Teaching Error Classes. In addition to "Error Not Listed" and "No Error" classes, the FM teaching error classes include:

(1) *Incorrect Demonstration of Skill:* A demonstration that fails to show how the intended skill should be performed. This type of demonstration does not support learning the correct execution of the skill.
(2) *Inefficient Demonstration:* A demonstration that accomplishes the task but does so using unnecessary motions. The intended goal state is reached, but it is just heavily inefficient.

The way in which we describe each of the FM teaching feedback error classes in the prompt can be found in Figure 3.

5.2 Domain Knowledge

The domain knowledge provided in the prompts is described in Figure 4.

5.3 Textual Prompts

This section contains the textual prompts for the FM feedback pipeline. For the FM planning feedback, this includes prompts for obtaining the FM skill breakdown (Figure 5) and obtaining FM planning errors (Figure 6).

For the FM teaching feedback, this includes prompts for obtaining the initial and final image textual summaries (Figure 7) and for obtaining FM teaching errors (Figure 8).

5.4 Example FM Outputs

FM Planing Feedback: Example received by participant 4431 (Both) in the soil domain where the FM classified their list of skills "(1) Scoop Material (2) Pour Material" as "Overfit Skill." The feedback it provided is as follows:

Each of your action items combines multiple distinct robot motions into a single skill. For example, "Scoop Material" likely includes both moving to the bucket and scooping, while "Pour Material" likely includes both moving to the mixing bucket and dumping. To improve your breakdown, please split each combined action into separate items — one for each distinct robot motion (e.g., "move to pre-scoop location", "scoop", "move to pre-dump location", "dump"). This will help ensure each skill is reusable and generalizable.

"Missing Skill": A skill that is required to complete the task but is absent from the current set of available skills. Without this skill, the task cannot be successfully completed.
"Irrelevant Skill": This skill should not be used to accomplish the task. It does not align with the task's goals in the domain and will actively degrade performance by reducing efficiency or causing failure.
"Redundant Skill": This skill can be used to accomplish the task, but it is not necessary because other available skills already cover its functionality. Including it does not harm task success or efficiency — it's just extra.
"Overly-General Skill": A skill that is too abstract to be effective. It lacks the necessary specificity to contribute meaningfully to task completion (e.g., overly vague motions like "move left" with no goal or context).
"Overfit Skill": A skill that is so narrowly defined that it only works in very specific situations. It cannot generalize to small variations of the task, such as changes in object number or layout. Often involves multiple valid skills taught at once rather than split up (e.g., "pick up hammer and go to nail and hit nail" treated as a single skill item instead of "pick up hammer", "go to nail", "hit nail" as three).
"Ambiguous or Misleading Action Labeling": A skill description that contains misleading phrasing or ambiguous object references, resulting in potential misinterpretation of the intended action. This can prevent the robot from learning the correct behavior even if the underlying motion is correct.
"Uncategorized Error": Something is wrong with the human action breakdown, but it doesn't align with errors types.

Figure 2: Error class descriptions for the LLM to employ for skill breakdown.

"Incorrect Demonstration of Action": One or more of the demonstrations fail to show how the intended action should be performed. This type of demonstration does not support learning the correct execution of the action. This can happen (1) if the goal item chosen by the human doesn't make sense for the action, for example trying to "turn water" or "grasp air", or (2) if the action itself is unsuccessful, such as turning a knob without actually rotating it or attempting to grasp an object but failing to pick it up.
"Inefficient Demonstration": A demonstration that accomplishes the task but does so using unnecessary motions. The intended goal state is reached, but it's just heavily inefficient.
"Uncategorized Error": The demonstration contains an error that does not fall into any of the predefined categories.

Figure 3: Error class descriptions for demo feedback.

FM Teaching Feedback: Example received by participant 2254 (FM) in the table setting domain where the FM classified their demonstration of the "pick_up_fork_1" skill as "Incorrect Demonstration of Action." The feedback it provided is as follows:

The robot arm did not move toward Fork 1 or perform a grasping action. Instead, it remained near the center of the table and ended above a neutral platform. To improve this demonstration, ensure the robot arm moves directly toward Fork 1 in the forks tray, aligns with it, and closes the gripper to pick it up. The action should clearly show the fork being grasped and lifted from its original position.

Domain Objects: Four identically shaped buckets are placed in front of the robot: a bucket of manure, a bucket of sand, a bucket of lime, and a mixing bucket.
Tracked Objects: Manure Bucket, Sand Bucket, Lime Bucket, Mixing Bucket
Participants are asked to manipulate the robot to create different soil mixtures for different plants.
In this domain, the robot's gripper is initialized to hold a scoop, and will always hold the scoop. The robot is able to scoop, dump, and move.

(a) Domain Knowledge for Soil Mixing.

Some utensils are placed in front of the robot. Two plates are placed on the table at different locations. Forks and knives are identical in shape and graspable the same way.
Tracked Objects: Fork 1, Fork 2, Knife 1, Knife 2, Plate 1, Plate 2
Participants are asked to manipulate the robot arm to set the table by picking up the utensils, and placing them in designated locations around two place settings.
The placement locations (where the utensils should be placed with respect to the plate) are not tracked by april tags.
However, the utensils should either be placed to the left of the plate or to the right of the plate. Forks and knives can be placed in either side.
The robot arm is able to move, open gripper, and close gripper.

(b) Domain Knowledge for Table Setting.

Figure 4: Domain knowledge for the two domains.

You are tasked with helping human demonstrators identify all the elementary actions in a domain for training robots in a Learning from Demonstration (LfD) setting.
Demonstrators may not be familiar with machine learning, so review the possible actions and suggest improvements by ensuring they:
- Avoid overgeneralizing actions when distinct control mechanisms are necessary for task success. - Generalize actions that use the same low-level control (e.g., 'pick red ball' and 'pick orange ball' should both be categorized under 'pick'). - Are flexible enough to generalize across similar tasks. - Are abstract enough to apply to related tasks but not so abstract that the goal is unclear (e.g., 'Go to green ball' rather than 'Move left/right'). - Do not include mid-motions unless they correspond to a distinct, goal-directed control step that is not part of the DMP trajectory between known start and end poses.
We are employing Dynamic Movement Primitives for the learning, so the actions learned should be able to generalize to different starting and stopping action locations. As such, actions that have the same movement and only differ in their item should be able to generalize. If the objects are (1) tracked in the environment and (2) share the same shape, then action generalization should be automatic, even without a demonstration for each object.
If the objects are not tracked, then an action cannot generalize to them and each should be learned independently.
Tracked objects can serve as the target of an action. If tracked objects are identical in shape, there should not exist multiple actions for them separately.
According to the specific domain setting, **if and only if** the robot gripper is able to open and close, you should include 'open gripper' and 'close gripper' as the elementary actions because they are pre-defined robot functions.
If you believe an action would include 'open gripper' or 'close gripper', do not include the gripper motion itself in the name of the elementary action. For example, when grabbing a stick, the gripper closes during the process. The step just before that should not be called 'grab'. Instead, you could name it something like 'move to pre-grab location', so that this step together with 'close gripper' clearly represents the full action of grabbing the stick.
Avoid including return-to-home motions unless they represent task-critical goals with a definitive goal position.
Domain: <insert-target-domain-name>
Domain knowledge: <insert-target-domain-knowledge>
Domain Tasks: <insert-target-task-0> <insert-target-task-1>
Only specify individual object or target differences in the action name if they influence the robot's movement or grasping behavior. Actions should be combined into one if the shapes of the target objects are identical (suggesting that required grasping actions are identical). For example, if a remote control and a smartphone are both flat, rectangular, and grasped in the same way, then "move to pre-grasp remote" and "move to pre-grasp phone" should be merged into "move to pre-grasp device".
Provide the actions and items in the target domain strictly adhering to this format, without anything redundant:
'[list of all possible elementary actions]' 'elementary action: [all possible tracked objects]'
Think first within the <think> <
think> tags, and check whether any your proposed actions include mid-motions not corresponding to a distinct, control step with a specified definitive goal position. If so, exclude them and revise. Then answer within the <answer> <
answer> tags. You may begin: <think>

Figure 5: Textual prompt for LLM composing optimal skill breakdown.

You are tasked with helping human demonstrators identify all the elementary actions in a domain for training robots in a Learning from Demonstration (LfD) setting. You've already determined the full set of elementary actions for the <insert-target-domain-name> domain. Your next step is to review the set of actions provided by a human demonstrator, and identify potential errors.
Domain knowledge: <insert-target-domain-knowledge>
Tasks for this example domain: <insert-target-task-0> <insert-target-task-1> The above tasks should be able to achieved by a permutation of elementary actions.
Optimal action breakdown you determined: <insert-llm-action-breakdown>
Human Action Breakdown: <insert-human-action-breakdown>
First, evaluate the Human Action Breakdown and the Optimal action breakdown you determined. You should note that both action breakdowns are unordered and should not be judged by the sequence of actions. Differences in action sequence are also allowed, as long as the actions can be aligned one-to-one after reordering. You should only judge whether all necessary action are present. Do not penalize missing repetitions needed for executing full tasks. You should not judge an action based on ANY context, just itself. If the two breakdowns can be unorderedly, one-to-one aligned with each other, respond with:
No Error
Important Clarifications:
Each action only needs to appear once in the breakdown to be considered available for reuse. You should allow difference in wording and terminology ambiguity as long as the meanings of the actions align. For example, "touch cup A" and "contact cup A (move the robot arm to touch it)" require the same physical motion and should be considered aligned. Individual actions can be used multiple times during task execution. Both action breakdowns are supposed to be a set of primitive actions that humans can freely choose and combine for a task execution. You should not infer whether an action will succeed based on its surrounding context or other actions. Evaluate each action only by its name and intended meaning. If an action is present in the set, it is assumed that it can be combined with other actions freely. The purpose of the evaluation is only to ensure that the complete set of necessary action types is available, NOT whether they are combined correctly.
In the Human Action Breakdown, each action item (e.g., "Action 1: ...", "Action 2: ...") must correspond to exactly one elementary action. If a single action item contains multiple actions connected by "and", even if the content matches the optimal breakdown, it must be separated into multiple action items. If you detect such cases, prioritize instructing the human to separate the actions into individual action items before considering any other feedback. You must flag any action item that contains multiple actions joined by "and", even if the individual actions are correct.
After taking those clarifications into account, if the two breakdowns can be aligned with each other, respond with:
No Error
By "the two breakdowns can be aligned with each other", I mean that each action in the human breakdown must align to one and only one action in the optimal breakdown, and vice versa. You must not align multiple human actions to a single optimal action, or multiple optimal actions to a single human action. The goal is to verify that the human set contains the correct types of primitive actions — not repeated or decomposed variants of the same action.
Otherwise, if the human breakdown cannot be aligned unorderedly, one-to-one (not n-to-one) with the optimal breakdown, pick from one of the errors below. One-to-one alignment means that each action item in the Human Action Breakdown must correspond to exactly one distinct elementary action in the Optimal Breakdown, and each elementary action in the Optimal Breakdown must be covered by exactly one human action item. If multiple human action items map to the same optimal action (many-to-one), or if one human action item maps to multiple optimal actions (one-to-many), you must flag it as an error. Importantly, even if repetitions are permitted for real-world execution, each individual action item must still uniquely correspond to a single primitive action when evaluating the breakdown. Coverage alone is not sufficient; strict one-to-one correspondence is required for this judgment.
<insert-error-skills>
You should select an error if applicable, and explain to the human what's wrong with their breakdown and how to improve it. If many-to-one or one-to-many alignments are detected, keep that in mind when you choose from the list of valid errors. Your answer should be structured as <error> "... Error" (must be an existing one in the error list) <error> <answer> ... <answer>
The "Explanation" in the answer should not be an explanation of your error choice. Instead, it will be shown directly to a human participant to highlight the issue and suggest how to improve. Please write it with that audience and purpose in mind.
If you pick "No Error", ONLY return "No Error" in both the <error><error> tags and <answer><answer> tags and nothing else.
Think first in the <think> <
think> tags. Your thinking should include aligning human action breakdown and the optimal action breakdown. After you made the alignment, discuss whether you spotted any many-to-one or one-to-many alignment before proceeding to the error decision. Then respond adhereing to the answer format within the <error><error> and <answer><answer> tags. You may begin: <think>

Figure 6: Textual prompt for skill breakdown feedback.

You're a helpful assistant. You are tasked with observing an image and provide detailed descriptions. The image will contain a robot arm and some other objects.
Your response should follow this process:
First, describe what identifiable objects you observe in the environment. Identifiable objects refers to specific items in the scene (e.g., orange, banana, cup), not general elements like the table, floor, or wall. Then, describe the position of the robot arm's end effector according to every one of them.
You must explicitly outline the spatial relation of the end effector with EVERY identifiable item you spotted in the image. If the items have text on them, associate it with their visual attribute and their spacial position. If the items have an specific order or relation, describe the order or relation you noticed. You should specifically relate the position of the robot arm end-effector to identifiable objects in the scene.

Figure 7: Textual prompt for demonstration initial and final image summary.

You're a helpful assistant. You are tasked with evaluating human-provided kinesthetic demonstrations for a robot learning-from-demonstration (LfD) setting. The domain and action set have been pre-defined. Each demo aims to represent a low-level, elementary action (e.g., "move the end effector to," "grasp") and must be demonstrated by the user.
Domain knowledge: <insert-domain-knowledge>
The current elementary action the user is demonstrating: <insert-skill-name>
The item associated with the current elementary action: <insert-item-name>
You're provided with the start and end frames of a user demonstration of that action.
In the start frame, you notice that: <insert-start-summary>
In the end frame, you notice that: <insert-end-summary>
You're also provided a visualization of the end effector's path for the demonstration trajectory, depicted from the camera's perspective. The paths are depicted with the Viridis palette, with yellow as the start position and blue as the end position, and labeled as: End-effector Trajectory
Your job is to decide whether this demonstration is a valid example of the intended elementary action.
Your task is to judge whether the human demonstration is appropriate for the target elementary action the user is demonstrating. If the demonstration perfectly aligns with the target action, respond with 'No Error'. If you identify potential errors, you should select from one of the categorized errors below. The end effector trajectory and the start/end frames should guide your choice.
For the end-effector trajectory, you should use it to infer how the arm moves between those states and describe it. Describe any inefficiency in the trajectory you see. All of the actions don't require obstacle avoidance.
Lastly, based on your observations, you should evaluate whether the demonstrated action is correct and efficient, and if it correctly attends to the specified item (only if item is not None). State any misalignment you noticed. The item associated with the current elementary action is: <insert-item-name>.
A set of possible error choices is listed below. If you believe the human demonstration is erroneous, select one from them and provide it as your answer:
<insert-error-demo>
You should select an error if applicable, and explain to the human what's wrong with their breakdown and how to improve it. Your answer should be structured as <error> "... Error" (must be an existing one in the error list) <error> <answer> ... <answer>
The "Explanation" in the answer should not be an explanation of your error choice. Instead, it will be shown directly to a human participant to highlight the issue and suggest how to improve. Please write it with that audience and purpose in mind.
If you pick "No Error", ONLY return "No Error" in both the <error><error> tags and <answer><answer> tags and nothing else.
Think first in the <think> <think> tags. Then respond adhereing to the answer format within the <error><error> and <answer><answer> tags. You may begin: <think>

Figure 8: Textual prompt for FM teaching feedback.

References

[1] Nakul Gopalan, Nina Moorman, Manisha Natarajan, and Matthew C Gombolay. 2022. Negative Result for Learning from Demonstration: Challenges for End-Users Teaching Robots with Task And Motion Planning Abstractions.. In *Robotics: Science and Systems*.

[2] Michael Kaiser, Holger Friedrich, and Rudiger Dillmann. 1995. Obtaining good performance from a bad teacher. In *Programming by Demonstration vs. Learning from Examples Workshop at ML*, Vol. 95. Citeseer.

[3] Nina Marie Moorman, Nakul Gopalan, Aman Singh, Erin Hedlund-Botti, Mariah L Schrum, Chuxuan Yang, Lakshmi Seelam, and Matthew Gombolay. 2023. Investigating the impact of experience on a user's ability to perform hierarchical abstraction. In *Robotics: Science and Systems*.

Received 2025-09-30; accepted 2025-12-23