

Robot Explanations: Preferences, Generation, and Communication

BY

Zhao Han

B.S. UNIVERSITY OF MANITOBA, CANADA (2014)

M.S. UNIVERSITY OF MANITOBA, CANADA (2016)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

UNIVERSITY OF MASSACHUSETTS LOWELL

AUGUST 2021

© 2021 by Zhao Han
All rights reserved

Robot Explanations: Preferences, Generation, and Communication

BY

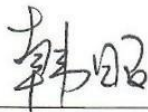
Zhao Han

B.S. UNIVERSITY OF MANITOBA, CANADA (2014)

M.S. UNIVERSITY OF MANITOBA, CANADA (2016)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
COMPUTER SCIENCE
UNIVERSITY OF MASSACHUSETTS LOWELL

Signature of
Author: _____



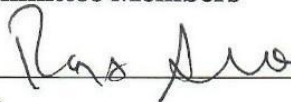
Date: July 7, 2021

Signature of Dissertation Chair:
Name Typed: Dr. Holly A. Yanco



Signatures of Other Dissertation Committee Members

Committee Member Signature:
Name Typed: Dr. Reza Ahmadzadeh



Committee Member Signature:
Name Typed: Dr. Aaron Steinfeld



Robot Explanations: Preferences, Generation, and Communication

BY

Zhao Han

ABSTRACT OF A DISSERTATION SUBMITTED TO THE
DEPARTMENT OF COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF MASSACHUSETTS LOWELL

AUGUST 2021

Dissertation Supervisor: Dr. Holly Yanco
Professor, Computer Science

Abstract

During the last decade, robots have become increasingly ubiquitous. They have been moved outside of laboratories and deployed to environments where they have to interact with humans. Examples include public places such as warehouses, hotels, factories, retail stores, and streets, as well as the most anticipated places – private homes. In these increasingly unstructured environments, requirements for human-robot interaction and collaboration are more involved as the tasks that robots complete are increasingly complex. For people to build trust with robots, there is a pressing need for robots to explain their actions and behaviors explicitly, rather than in implicit and vague manners such as using eye gaze or arm movement. This dissertation centered around robot explanations and examined four interconnected aspects of the robot explanation process: from what explanations humans prefer, how to generate explanations, and how to communicate them explicitly, to explaining missing causal information of past actions due to environment change.

The contributions of this work are fourfold. In a human-subjects study, we found strong evidence that people prefer verbal explanations coupled with non-verbal cues. To verbally explain, people prefer robots to get their attention first, then concisely explain, and are only willing to ask a few follow-up questions for more details. Then I contributed explanation generation algorithms using Behavior Trees (BTs), a simple yet powerful robot task sequence method for high-level and failure robot explanations. We framed BTs into semantic sets to generate explanations from the resulted shallow tree and demonstrated the algorithms with a complex mobile manipulation task and a taxi domain navigation task. BTs were also made dynamically modifiable for behavior insertion after users' follow-up questions. Thirdly, we contributed a complete projection mapping implementation solution for instant and salient robot communication: from how to choose an off-the-shelf projector, how to calibrate it, and the underlying principle, to all the code and files needed to readily integrate the solution into any ROS system. Finally, I investigated physical replays, verbal and projection markers for robots to help people infer missing causal information

of a robot's past actions. We found that a multimodal approach, including all physical replay, verbal, and projection markers, provided better aid in inference-making, less mental workload, and more trustworthy robots.

Acknowledgements

All the work has been supported in part by the Office of Naval Research (N00014-18-1-2503) and the National Science Foundation (IIS-1763469). During my doctoral study, I had the opportunity to closely supervise some undergraduate students working as interns in the human-robot interaction laboratory at UMass Lowell. I have also collaborated with other universities Tufts and CMU (Carnegie Mellon University) under the Office of Naval Research grant. Their work is acknowledged in each chapter if applicable.

First, I would like to thank my advisor, Prof. Holly Yanco, for providing guidance and resources throughout these years. When I started my Ph.D., human-robot interaction or robotics was a brand-new area to me because I had studied data mining during my M.S. studies. It is her who makes me a confident roboticist and a better researcher. Prof. Yanco has always provided thoughtful suggestions and invaluable edits to the papers we publish together. She also allowed me to pursue research freely with encouragement and made the learning journey enjoyable and worthwhile. Besides, her insights into life from time to time during our weekly meetings and other meetings also made me a better person. Concisely, without her tremendous support, this dissertation would not be possible.

I would also like to thank my other committee members, Prof. Reza Ahmadzadeh and Prof. Aaron Steinfeld, for their willingness to put effort into this dissertation. I want to thank Prof. Ahmadzadeh for allowing us to use the Fetch robot and set up a test course for the mobile manipulation task used as a research platform. Thank you, Prof. Aaron Steinfeld, for collaborating with Prof. Yanco on the grant that ultimately supports my research.

This dissertation would also not be the same without the help from people in the human-robot interaction lab and at NERVE¹, especially our lab manager – Jordan Allspaw, my labmates – Abraham Shultz, Gregory LeMasurier, and James Kuczynski, our NERVE staff – Adam Norton

¹Directed by Prof. Holly Yanco, the New England Robotics Validation and Experimentation (NERVE) Center is an interdisciplinary robotics testing, research, and training facility.

and Brian Flynn, and our undergraduate students – Daniel Giger, Jenna Parrillo, Alexander Wilkin-son, and Patrick Hoey. Thank you all for proofreading my paper drafts, listening to my practice talks, providing hardware support, helping to compete in the robotic competitions, and improving the resulted work, which are all incorporated in this dissertation.

Last but not least, I'd like to express my gratitude to my parents, my wife, and my little son. Without my parents' support, I would not be able to come to a foreign country and peruse a career in robotics. Without my wife and our lovely son, I would lose so many wondering moments that support me to do research on the following mornings.

There are not many five years in one's life. I have enjoyed being part of the lab and will continue to appreciate being a husband and a father.

Table of Contents

| | |
|--|-------------|
| Abstract | i |
| Acknowledgements | iii |
| Table of Contents | v |
| Publications | xi |
| List of Tables | xii |
| List of Illustrations | xiii |
| List of Algorithms | xxiv |
| 1 Introduction | 1 |
| 1.1 Research Questions | 2 |
| 1.2 Statement of Problem | 3 |
| 1.3 Approach | 3 |
| 1.4 Contributions | 6 |
| 2 Related Work | 9 |
| 2.1 Desired Explanations | 9 |
| 2.2 State Summarization | 11 |
| 2.2.1 Manual Methods | 11 |
| 2.2.2 Summarization Algorithms | 12 |
| 2.3 Robot Task Representation: Why Behavior Trees for Robot Explanations | 14 |
| 2.4 Robotic Data Storage and Querying | 17 |
| 2.4.1 Storing Unprocessed Data | 17 |

| | | |
|----------|--|-----------|
| 2.4.2 | Storing Processed Data | 19 |
| 2.4.3 | Querying | 20 |
| 2.5 | Human Interface for Communication | 22 |
| 2.5.1 | Display Screen | 23 |
| 2.5.2 | Augmented Reality (AR) | 24 |
| 2.5.3 | Robot Activities | 25 |
| 2.6 | Observational Learning | 26 |
| 3 | Desired Robot Explanation | 29 |
| 3.1 | Introduction | 29 |
| 3.2 | Hypotheses | 31 |
| 3.2.1 | The Need For Robot Explanations | 32 |
| 3.2.2 | Expected Properties of Robot Explanations | 32 |
| 3.3 | Method | 33 |
| 3.3.1 | Power Analysis, Participants, and Participant Recruitment | 33 |
| 3.3.2 | Robot Platform | 35 |
| 3.3.3 | Measures | 35 |
| 3.3.4 | Experimental Design | 37 |
| 3.3.5 | Procedure | 41 |
| 3.4 | Results | 42 |
| 3.4.1 | H3.1: Robot behavior will be considered unexpected and needs to be explained | 42 |
| 3.4.2 | H3.1, H3.2 & H3.3: Causal information, Headshake, and need for explanations | 44 |
| 3.4.3 | H3.4: Explanation timing | 47 |
| 3.4.4 | H3.5: Engagement importance/preference | 48 |

| | | |
|----------|--|-----------|
| 3.4.5 | H3.6: Similarity to human explanation | 49 |
| 3.4.6 | H3.7, H3.8: Detail, summarization and follow up questions | 50 |
| 3.4.7 | Additional analyses: Explanation content | 52 |
| 3.4.8 | Additional analyses: Reasoning behind explanation content | 54 |
| 3.5 | Replicating The Study To Verify Results | 57 |
| 3.5.1 | Additional analyses: Explanation content | 59 |
| 3.5.2 | Additional analyses: Reasoning behind explanation content | 60 |
| 3.6 | Discussion | 61 |
| 3.7 | Limitations and Future Work | 65 |
| 3.8 | Conclusions | 66 |
| 4 | Explanation Generation Using Behavior Trees | 68 |
| 4.1 | Introduction | 68 |
| 4.2 | Background: Formulation of Behavior Trees | 70 |
| 4.3 | Modeling Robotic Tasks using Behavior Trees | 72 |
| 4.3.1 | The Gearbox Kitting Task | 72 |
| 4.3.2 | Revised Notation for Behavior Trees | 75 |
| 4.3.3 | Modeling Using Behavior Trees | 75 |
| 4.3.4 | The Screw Picking Subtask: Tree Breakdown | 80 |
| 4.4 | Framing Behavior Trees for Hierarchical Explanation Generation | 81 |
| 4.5 | Algorithms on Behavior Trees for Robot Explanation Generation | 83 |
| 4.5.1 | Supporting Querying Current State | 84 |
| 4.5.2 | Supporting Hierarchical Explanation Generation | 85 |
| 4.5.3 | Supporting Failure Explanation Generation | 86 |
| 4.5.4 | Supporting Dynamic Behavior Insertion as Subgoal | 88 |
| 4.6 | Case Studies | 92 |

| | | |
|----------|---|------------|
| 4.6.1 | Large Gear Insertion: A Machining Task | 92 |
| 4.6.2 | Hierarchical Explanation Generation | 92 |
| 4.6.3 | Dynamic Behavior Insertion as Subgoal | 95 |
| 4.6.4 | Explanations of Failures | 96 |
| 4.6.5 | Taxi Domain: A Navigation Task | 97 |
| 4.6.6 | Explaining Divergences Between Behaviors | 101 |
| 4.7 | Limitations and Future Work | 103 |
| 4.8 | Conclusion | 105 |
| 5 | Projection Mapping Implementation | 106 |
| 5.1 | Introduction | 106 |
| 5.2 | Overview | 108 |
| 5.3 | Projector Selection Consideration | 108 |
| 5.4 | Projector Calibration | 109 |
| 5.5 | Virtual Camera | 111 |
| 5.6 | Projection Output | 111 |
| 5.7 | Hardware Platform | 112 |
| 5.8 | Conclusion | 113 |
| 6 | Communicating Missing Causal Information of Past Actions | 114 |
| 6.1 | Introduction | 114 |
| 6.2 | Hypotheses | 118 |
| 6.3 | Experiment Design | 119 |
| 6.3.1 | Task | 119 |
| 6.3.2 | Study Conditions | 120 |
| 6.3.3 | Questionnaire | 125 |
| 6.3.4 | Quality Assurance Questions | 129 |

| | | |
|----------|--|------------|
| 6.3.5 | Procedures | 129 |
| 6.3.6 | Power Analysis, Participants, and Participants Recruitment | 130 |
| 6.3.7 | Implementation | 132 |
| 6.4 | Results | 132 |
| 6.4.1 | H6.1. Effective causal inference with verbal markers (partial support) . . . | 133 |
| 6.4.2 | H6.2. Effective causal inference with projection markers (partial support) . | 137 |
| 6.4.3 | H3 – Efficiency. Faster causal inference with projection markers (partial support) | 137 |
| 6.4.4 | H3 – Accuracy. More accurate causal inference with projection markers (not supported) | 145 |
| 6.4.5 | H6.4. The same workload in both verbal and projection conditions (mostly supported) | 150 |
| 6.4.6 | H6.5. A robot is more trustworthy with projection markers (not supported) | 152 |
| 6.4.7 | H6.6. Less workload when presented both verbal and projection markers (almost not supported) | 155 |
| 6.5 | Discussion and Recommendations | 156 |
| 6.6 | Effectiveness of inferring missing causal information from the past | 156 |
| 6.7 | Efficiency to infer past missing causal information | 158 |
| 6.8 | Inference accuracy and confidence of past missing causal information | 159 |
| 6.9 | Mental workload for past causal information inference | 161 |
| 6.10 | Perceived trust as a result of causal inference indications | 161 |
| 6.11 | Limitations and Future Work | 161 |
| 6.12 | Conclusion | 163 |
| 7 | Conclusions and Future Work | 164 |
| 7.1 | Conclusions and Contributions | 164 |

| | | |
|-------|--|------------|
| 7.2 | Future Work | 164 |
| | Literature Cited | 167 |
| | Appendix A Full Results from the Studies in Desired Robot Explanation | 182 |
| A.1 | Internal Consistency (2019 Study vs 2020 Study) | 182 |
| A.2 | Unexpectedness | 183 |
| A.2.1 | Unexpectedness (interaction effect): No change | 183 |
| A.2.2 | Unexpectedness (bar chart) | 185 |
| A.2.3 | Unexpectedness (box plot) | 187 |
| A.3 | Need (Question 1): Same conclusions | 188 |
| A.3.1 | Results for 2019 Study (Figure 70) | 188 |
| A.3.2 | Results for 2020 Study (Figure 71) | 190 |
| A.3.3 | Results for both experiments combined (Figure 72) | 191 |
| A.4 | Need: Statistical significance remains the same | 193 |
| A.4.1 | Need (interaction effect): No change, still no interaction effect | 193 |
| A.4.2 | Need (bar chart): Same Conclusions | 195 |
| A.4.3 | Need (box plot) | 197 |
| A.5 | Expected properties | 198 |
| A.5.1 | Explanation timing and verbosity preferences: Conclusions remain the same | 198 |
| A.5.2 | Engagement importance/preference: Conclusions remain the same | 199 |
| A.5.3 | Similarity to human explanation: Largely remained the same; the same conclusion | 200 |
| A.5.4 | Detail, summarization: Same conclusions | 201 |

Publications

Portions of this dissertation have resulted in the following peer-reviewed conference and journal papers:

1. **Chapter 2:** Zhao Han, Jordan Allspaw, Adam Norton, and Holly A. Yanco. 2019. Towards A Robot Explanation System: A Survey and Our Approach to State Summarization, Storage and Querying, and Human Interface. In *Proceedings of The Artificial Intelligence for Human-Robot Interaction (AI-HRI) Symposium at AAI Fall Symposium Series (AAAI-FSS) 2019*.
2. **Chapter 3:** Zhao Han and Holly A. Yanco. 2020. Reasons People Want Explanations After Unrecoverable Pre-Handover Failures. In *ICRA 2020 Workshop on Human-Robot Handovers*.
3. **Chapter 3:** Zhao Han, Elizabeth Phillips, and Holly A. Yanco. 2021. The Need for Verbal Robot Explanations and How People Would Like a Robot To Explain Itself. Accepted to *ACM Transactions on Human-Robot Interaction (THRI)*, awaiting publication.
4. **Chapter 4:** Zhao Han, Daniel Giger, Jordan Allspaw, Michael S. Lee, Henny Admoni, and Holly A. Yanco. 2020. Building The Foundation of Robot Explanation Generation Using Behavior Trees, 10 (3). *ACM Transactions on Human-Robot Interaction (THRI)*.
5. **Chapter 5:** Zhao Han, Alexander Wilkinson, Jenna Parrillo, Jordan Allspaw and Holly A. Yanco. 2021. Projection Mapping Implementation: Enabling Direct Externalization of Perception Results and Action Intent to Improve Robot Explainability. In *Proceedings of The Artificial Intelligence for Human- Robot Interaction (AI-HRI) Symposium at AAI Fall Symposium Series (AAAI-FSS) 2020*.
6. Zhao Han, Jordan Allspaw, Gregory LeMasurier, Jenna Parrillo, Daniel Giger, S. Reza Ahmadzadeh and Holly A. Yanco. 2020. Towards Mobile Multi-Task Manipulation in a Confined and Integrated Environment with Irregular Objects. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*.

The following manuscript is currently *under review* as of August 2021:

1. **Chapter 6:** Zhao Han and Holly A. Yanco. 2021. Explaining a Robot’s Past: Communicating Missing Causal Information by Replay, Speech and Projection. *Submitted to ACM Transactions on Human-Robot Interaction (THRI)*.

List of Tables

| | | |
|---|---|-----|
| 1 | Explanation Measure Items | 35 |
| 2 | Study Conditions Across The Two Factors | 39 |
| 3 | Summary of evidence or lack thereof for hypotheses (includes data from both the 2019 Study and the 2020 Study) | 63 |
| 4 | Classical Behavior Tree (BT) Formulation: Nodes and Return Statuses. (Adapted from [40]) | 72 |
| 5 | Notation of and changes to Behavior Trees as used in this chapter | 79 |
| 6 | Questions To Be Answered During Hierarchical Explanation | 85 |
| 7 | Causal Verbal Markers And Their Timing | 123 |
| 8 | The effectiveness and efficiency of the conditions in inference-making. Shaded boxes are the best values in each column. | 160 |
| 9 | Explanation Measure Items | 182 |

List of Illustrations

| | | |
|---|--|----|
| 1 | Three of the six handover conditions after the almost-reachable cup is detected, without the headshake included. <i>Left</i> : Robot does nothing (No Cue). <i>Middle</i> : Robot’s head turns towards the cup (Look). <i>Right</i> : Robot’s head turns towards the cup and its right arm is extended repeatedly (Look & Point). | 30 |
| 2 | To establish the handover task context, the image above was first presented to participants with the three yellow pop-ups slowly fading in one after another clockwise from the top left. | 38 |
| 3 | The distribution of Unexpectedness responses with median lines and estimated marginal means. Except for Look without Headshake and Look & Point without Headshake, all conditions were rated unexpected (first two lines of the annotation in each of the boxes, comparison against 0). Results of post-hoc pairwise comparisons are also shown (second and third lines of the annotation in each of the boxes). | 43 |
| 4 | Boxplot of Unexpectedness responses. All robot behaviors were rated unexpected except for Look without Headshake (the middle red box) and Look & Point without Headshake (the right red box). | 44 |
| 5 | The distribution of Need responses with median lines and estimated marginal means, showing that robot behavior should be explained in all conditions. No statistical significance was found in post-hoc pairwise comparisons. | 45 |
| 6 | Boxplot of the Need for explanation responses. Participants in all conditions agreed that robot behavior should be explained. No significant differences were found pairwise between conditions. | 46 |

| | | |
|----|---|----|
| 7 | Timing and verbosity preferences. Around half participants prefer the robot to explain <i>in situ</i> and most (75%) participants are willing to ask only a few clarifying follow-up questions. | 47 |
| 8 | Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). Most participants (69%) agree it is important for the robot to engage with them before explaining. Slightly over one-third of all participants prefer looking at them to get their attention. | 48 |
| 9 | Robot vs. human explanations in what and how (green line indicates median). Half participants agreed on no differences in both. Please see Section 3.4.5 for more details. | 49 |
| 10 | Two summarization aspects (green lines indicate median values). <i>Detailed</i> : While 35% participants (1 & 3) agreed explanations should be detailed, almost half of the responses (-2, -1, 0, 2) may happen at random. <i>Summarized</i> : 72% participants preferred explanations to be concise. | 50 |
| 11 | The most wanted robot explanations (Top categories regarding explanation content). The top one is why the robot failed to hand the cup. | 51 |
| 12 | The most wanted robot explanations across conditions (Most frequent categories regarding explanation content across conditions). An explanation for why the robot failed to pass the cup is common in all conditions. Please see Section 3.4.7 for other categories. | 52 |
| 13 | Most frequent coded responses for why participants want the robot to explain (i.e., explanation reasoning). Two of them are endorsed by more than 20% of participants: the robot should explain because its behavior does not meet their expectations, and in order to confirm the robot will do the task, will do it correctly, and whether it is capable of finishing the task. | 54 |

| | | |
|----|---|----|
| 14 | Top codes for explanation reasoning across conditions. Please see Section 3.4.8 for a detailed analysis. | 55 |
| 15 | Top categories regarding explanation content (replication results from the 2020 Study). The top one remains unchanged. | 58 |
| 16 | Most frequent categories regarding explanation content across conditions (replication results from the 2020 Study). Still, the explanation for why the robot failed to pass the cup is common in all conditions. Please see Section 3.5.1 for other categories. | 59 |
| 17 | Most frequent coded responses for explanation reasoning (replication results from the 2020 Study). The top two remain unchanged and are still endorsed by more than 20% participants. For the changes from the 2019 Study, please see Section 3.5.2. | 61 |
| 18 | Top codes for explanation reasoning across conditions (replication results from the 2020 Study). please see Section 3.5.2 for a detailed analysis. | 62 |
| 19 | Parts to be collected: (a) Large gear (b) Gearbox top (c) Gearbox bottom (d) Screw (e) Small gear. Note that the large gear (a) is meant to be machined to have threads; the large gear must be inserted into a machine for this process to occur. | 73 |
| 20 | Assembled gearbox using the required mechanical parts that the robot placed into the caddy and delivered to the inspection table. | 73 |
| 21 | The arena where the gearbox kitting task is carried out by a Fetch robot. Rendered in Gazebo, the main goal is to place a specified set of parts into the correct sections of the caddy, then to transport the caddy to the inspection table. | 74 |

| | | |
|----|--|----|
| 22 | The top level of the gearbox kitting task represented as a sequence of subtrees in a Behavior Tree. For readability, six tasks – go {pick — place} {small gear — gearbox top — gearbox bottom} – are represented by “...”. Note that the root node on the top is merely a pointer to the real root node in the middle, which is why depth 0 is at the sequence node. | 76 |
| 23 | The representative screw picking subtask modeled in Behavior Trees. The leftmost dot indicates that the fallback node has a parent, but the subtree parent node and other ancestor nodes are hidden here as they are shown in the previous figure. . . . | 77 |
| 24 | Screw placing, another representative subtask of the gearbox kitting task, modeled as a Behavior Tree. Similar to the previous figure, the leftmost node’s ancestors are not shown. For more detail, refer to Section 4.4. | 78 |
| 25 | Simplified, semantic sets for the screw picking subtask. The goal is not shown here as Figure 22 is sufficient without any simplification. | 82 |
| 26 | Simplified, semantic sets for the screw placing subtask. | 83 |
| 27 | The large gear insertion subtask modeled in Behavior Trees. | 93 |
| 28 | The framed BT for the large gear insertion subtask. | 94 |
| 29 | The behavior tree for the kitting task after answering ““Can you insert large gear?”. Most subgoals in Fig. 22 are collapsed for readability. The empty sequence node at depth 1 is dynamically inserted to ensure all input ports are satisfied by output ports. See Section 4.6.3 for more. | 96 |
| 30 | An environment of the taxi Domain, in which the agent delivers a passenger to the destination. Row and column numbers are added for easy reference. | 97 |
| 31 | The behavior tree representation of an optimal policy in the non-manipulation taxi domain. Given that it is less complex than a kitting subtask, there is no need for simplification. | 98 |

| | | |
|----|--|-----|
| 32 | An agent’s optimal policy vs. a humans’s expected policy for the environment in Fig. 30. Divergences in actions (some leaf nodes) between the two policies are colored in dark blue and their immediate parents (some sequence nodes) are colored in blue-gray. | 102 |
| 33 | The projection of perception results: the detected objects (white and green) and the object to be manipulated (green). Using our implementation of projection mapping, researchers and practitioners can enable a robot to accurately externalize internal states for explanation. A video is available at https://youtu.be/S0z9e2gUrEA | 106 |
| 34 | High-level diagram for our projection mapping implementation. With the projector lens calibrated, a virtual camera – placed in Rviz with the same pose as the projector in real world – subscribes to the camera intrinsics so it can output an image of objects visualized in the virtual world in Rviz to the projector to reflect the perceived objects. See Section 5.2 for more details. | 107 |
| 35 | We calculated the intrinsics of our particular projector by mounting it perpendicular to a posterboard at a fixed distance. See Section 5.4 for more details. | 109 |
| 36 | A sample use, where a projector is mounted onto a Fetch robot via a custom hardware structure attached to its upper back and a turret unit to pan and tilt the projector. However, the projection mapping technique is robot agnostic and the projector does not have to be attached to the robot. See Section 5.7 for more details. | 112 |
| 37 | A mobile manipulation task environment in which we investigated how could the robot provide indicators to past missing causal information. The robot is supposed to pick different gearbox parts, including the gearbox bottoms on the table it is facing, take them to the caddy table on the left, and deliver the caddy to the bottom right table for an assembly worker to assemble a gearbox. | 115 |

38 The failure scenario that inspired the first picking scenario: A Fetch robot misrec-
 ognized a torn up wood chip near the top-right corner of the table as a screw. 117

39 The original scene where the robot avoided a ground obstacle, the yellow wet floor
 sign, to navigate to the caddy table to the left. At replay time, the yellow wet floor
 sign is gone. Key videos frames for the replay video without the sign is shown in
 Figure 42. 117

40 Snapshots from the picking videos with physical replay. During the picking task,
 the robot mistreated a wood chip as a gearbox bottom on the right edge of the table.
 Participants were asked to infer where the robot has picked. Verbal indicators are
 in Table 7. Projection photos are in Figure 41. There was no arm movement in the
 Say, Project, and Project-Say conditions. 121

41 Tabletop projections in the picking videos. Recognized objects are projected in
 white; The target object to be picked is in green. These three snapshots are cor-
 responding to the first, third, and fourth snapshot in the previous figure (Figure
 40). 121

42 Snapshots from the navigation video in the Replay-Project condition, where partic-
 ipants need to infer where the ground obstacle (a wet floor sign) was. The yellow
 projection consists of multiple 3D spheres of point clouds from the robot’s base
 laser scan. Purple arrows indicate the robot’s detour path. Non-projection condi-
 tions had no projection on the ground. For conditions with speech, its text is shown
 in Table 7. In the Say, Project, and Project-Say conditons, the robot’s base did not
 move. 122

43 Snapshots from the placing video in the Replay condition, where participants need
 to infer which section of the caddy that the robot placed into. Again, for speech
 condition, the text is shown in Table 7. Projection photos are shown in the next
 figure (Figure 44). 122

44 Tabletop projections in the placing videos. The last snapshot is from the Replay condition video and is indented to easily differentiate the projection from the photos to its left. The first three snapshots are corresponding to the first, the second, and the fourth snapshot in the previous figure (Figure 43) The rightmost reference image has been provided here to show what the image looks like without any projection, to allow for differentiating where the projection is on the left three figures. 123

45 Photo shown to participants to answer where the robot picked. The correct answer is “F”. 125

46 To answer where the ground obstacle was, this photo was shown to participants. The correct answer is “Area D”. 126

47 To answer which section of caddy the robot placed into, this photo was shown to participants. The correct answer is “Section A”. 127

48 Manipulation inference responses. “F” is correct. Replay conditions perform the best: nearly all participants were correct. Half wrongly selected nearby E in Say. Project and Project-Say only have half participants correct. 134

49 Navigation inference responses. The correct answer is “Area D”. Most participants were correct in all conditions except for the Say condition, in which only 40% were correct and half selected nearby Area E. 134

50 Placement inference responses. “Section A” is the correct answer. Around 60% participants could infer correctly except for Project, in which no statistically significant results were found. 136

51 Participant’s responses to when they have inferred the picking location. The Say condition performs the best with 60+ correct participants but 20+ never know. Project and Project-Say are at the second tier with fewer correct participants and more unknowing participants. In all replay conditions, the top four in the figure, participants reported they know at a later event. 139

52 Participant’s responses to when they have inferred where the ground obstacle was in replay conditions. In summary, Replay-Project-Say performed the best, with 20+ participants inferring at the earliest event: before the robot started moving. (Non-replay conditions, excluding the Project condition where projection was always on, had their own options as the robot’s base did not move during these conditions; See Figure 53 and 54.) 141

53 Participant’s responses in the Say condition to when they have inferred where the ground obstacle was. Most participants made the inference after the robot started speaking, more than any option in the replay conditions shown in the previous figure (Figure 52). (The options were only present in the Say condition as the robot did not move its base but just spoke.) 142

54 Participant’s responses to when they have inferred where the ground obstacle was. More than 60% of participants made the inference from the ground projection. (The options were only present to the Project-Say condition as the robot did not move its base but made projection onto the ground and spoke.) 143

55 Participant’s responses to when they have inferred which section of the caddy the object was placed into. Participants in the Project condition made the earliest inference after the robot’s head stopped moving. However, 30+ participants reported they never knew. Say and Project-Say has the top performance because the participants elaborated on what they knew from the robot’s speech. For replay conditions, arm movement significantly delays the inference. 144

56 Participant’s confidence levels in their responses to the picking inference question. Generally, participants in replay conditions are more confident than those in replay conditions (Confident or very confident in replay conditions vs. somewhat confident in non-replay conditions). 146

| | | |
|----|---|-----|
| 57 | Participant’s confidence levels in their responses to the navigation inference question. Participants’ confidence levels in the Project and Project-Say conditions are increased to confident from somewhat confident in their picking inference. However, the statistical significances suggest that participants are still more confident in replay conditions (more right-skewed). | 147 |
| 58 | Participant’s confidence levels in their responses to the placement inference question. Participants in the Project condition has more unsure ratings, while other condition has more participants distributed in different confidence level ratings. . . | 149 |
| 59 | Responses to the NASA Task Load Index questionnaire. Results from pairwise comparisons are shown and dashed lines indicate median values. In general, replay conditions and the Say condition performed the best; No statistically significant differences were found between the Say and Project conditions except for performance. See Section 6.4.5 for more details. | 151 |
| 60 | Responses to the Muir trust questionnaire [120]. Regarding predictability, participants rated non-replay conditions less predictable. For reliability, when accompanied with replays or with both replays and verbal indicators, projection markers improve reliability. In terms of competence, adding replay to either Say or Project or both increases the competence rating. For the direct trust measure, replay conditions have more positive ratings than their non-replay counterparts. See Section 6.4.6 for more details. | 153 |
| 61 | Interaction plot of the unexpectedness responses (original result from the 2019 Study). | 183 |
| 62 | Interaction plot of the unexpectedness responses (replication result from the 2020 Study). | 183 |
| 63 | Interaction plot of the unexpectedness responses (combined result). | 183 |

| | | |
|----|--|-----|
| 64 | The distribution of Unexpectedness responses with median lines (original result from the 2019 Study). | 185 |
| 65 | The distribution of Unexpectedness responses with median lines (replication result from the 2020 Study). | 186 |
| 66 | The distribution of Unexpectedness responses with median lines (combined result). | 186 |
| 67 | Boxplot of Unexpectedness responses (original result from the 2019 Study). | 187 |
| 68 | Boxplot of Unexpectedness responses (replication result from the 2020 Study). | 187 |
| 69 | Boxplot of Unexpectedness responses (combined result). | 187 |
| 70 | The distribution of Need (questions 1) responses with median lines (original result from the 2019 Study). | 188 |
| 71 | The distribution of Need (questions 1) responses with median lines (replication result from the 2020 Study). | 189 |
| 72 | The distribution of Need (questions 1) responses with median lines (combined result). | 190 |
| 73 | Interaction plot of the unexpectedness responses (original result from the 2019 Study). | 193 |
| 74 | Interaction plot of the Need responses (replication result from the 2020 Study). | 193 |
| 75 | Interaction plot of the Need responses (combined result). | 193 |
| 76 | The distribution of Need responses with median lines (original result from the 2019 Study). | 195 |
| 77 | The distribution of Need responses with median lines (replication result from the 2020 Study). | 196 |
| 78 | The distribution of Need responses with median lines (combined result). | 196 |
| 79 | Boxplot of the Need for explanation responses (original result from the 2019 Study). | 197 |
| 80 | Boxplot of the Need for explanation responses (replication result from the 2020 Study). | 197 |
| 81 | Boxplot of the Need for explanation responses (combined result). | 197 |

| | | |
|----|---|-----|
| 82 | Timing and verbosity preferences (original result from the 2019 Study). | 198 |
| 83 | Timing and verbosity preferences (replication result from the 2020 Study). | 198 |
| 84 | Timing and verbosity preferences (combined result). | 198 |
| 85 | Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). [original result from the 2019 Study] | 199 |
| 86 | Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). [replication result from the 2020 Study] | 199 |
| 87 | Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). [combined result] | 199 |
| 88 | Robot vs. human explanations in what and how (green line indicates median). [original result from the 2019 Study] | 200 |
| 89 | Robot vs. human explanations in what and how (green line indicates median). [replication result from the 2020 Study] | 200 |
| 90 | Robot vs. human explanations in what and how (green line indicates median). [combined result] | 201 |
| 91 | Two summarization aspects (green lines indicate median values). [original result from the 2019 Study] | 201 |
| 92 | Two summarization aspects (green lines indicate median values). [replication result from the 2020 Study] | 202 |
| 93 | Two summarization aspects (green lines indicate median values). [combined result] | 202 |

List of Algorithms

| | | |
|---|--|----|
| 1 | Answer “What are you doing?” (Q1) | 84 |
| 2 | Answer “Why are you doing this?” (Q2) | 86 |
| 3 | Answer “What is your subgoal?” (Q3) | 86 |
| 4 | Answer “What is your goal?” (Q4) | 87 |
| 5 | Find Steps From a Goal or Subgoal Node | 87 |
| 6 | Answer “How do you achieve your { goal — subgoal } ?” (Q5) | 88 |
| 7 | Answer “Was there anything wrong?” “What went wrong?” “How was the failure handled?” | 89 |
| 8 | Find Self-Contained Behavior Node | 90 |
| 9 | Append Self-Contained Behavior Node As a Subgoal | 91 |

1 Introduction

As robots are being pushed by researchers and the industry to complete more complex tasks, improving the understanding of a robot's behaviors while it is completing the tasks is increasingly important. With the advancement and wide adoption of deep learning techniques, explainability of software systems and interpretability of machine learning models has attracted both human-computer interaction (HCI) researchers (e.g., [1]) and the artificial intelligence (AI) community (e.g., [114]). Work in human-robot interaction (HRI) has shown that improving understanding of a robot makes the robot more trustworthy [46] and the human-robot interaction (HRI) more efficient [7]. However, how robots can explain themselves at a holistic level, i.e., by generating explanations and communicating them, remains an open research question.

As opposed to virtual AI agents or computer software, robots have physical embodiment, which influences metrics such as empathy [146] and cooperation [14] with humans. Given this embodiment, some research in human-agent interaction does not apply to human-robot interaction. For example, in a literature review about explainable agents and robots [11], approximately half (47%) of the explanation systems examined used text-based communication methods, which is less relevant for robots that are not usually equipped with display screens. Instead, HRI researchers have been largely exploring non-verbal physical behavior such as arm movement [56, 100] and eye gaze [118]. Non-verbal behaviors can help people to anticipate a robot's actions [102], but understanding why that behavior occurred can further improve one's prediction of behaviors, especially if the behavior is opaque [108]. Thus, robot explanations of their own behavior are needed.

For humans, explaining our behavior is a natural part of daily life; the lack of explanations is unsettling and disturbing [71]. Psychologists have long studied human explanations. As categorized by Malle [108], we either explain what was unexpected or strange to have a complete and coherent understanding, known as meaning-finding explanations, or use explanations as communicative acts to create shared meaning and manage social interactions to influence the

explainee's mind or behavior, known as interaction-managing explanations. From an early age, people seek explanations for clarification and further understanding when surprised or confused [108, 90, 166]. As a counterexample, Moerman [117] found that patients feel better when they receive explanations about their illness. Thus, it is important for others to provide explanations to improve understanding. This need is equally true for robots, often designed as intelligent beings with physical appearances that resemble humans.

Yet to our best knowledge, it remains underexplored how robots should be enabled to explain, despite the large body of HRI research on non-verbal cues to indicate a robot's intent (e.g., [56, 118, 100]).

1.1 Research Questions

There are 3 major research questions addressed in this dissertation. They are listed below with subproblems so we can divide and conquer to achieve the goal. These research questions served as a guide pointing to the directions of this thesis.

1. **Desired robot explanation:** What do people want a robot to explain? Are verbal explanations necessary for a robot that already shows non-verbal motion cues? If so, how can we ensure that the robot explains itself in a way that is preferred by humans?
2. **Explanation generation:** How can a robot generate explanations from its internal states? What is the best representation to organize a robot's internal states? How can we enable roboticists to provide explanations through programming?
3. **Explanation communication:** Last but not least, how could robots communicate explanations? What other modalities can a robot combine to better communicate with people? For long-term human-robot interaction or post-hoc introspection, how can a robot explain its behavior and decisions in the past?

1.2 Statement of Problem

In my thesis, I wanted to investigate robot explanations desired by humans, develop techniques to generate and communicate explanations. Human subjects studies were planned for evaluation.

Specifically, to enable a robot to explain itself, robot designers must have insight into what and how humans want robots to explain its behavior, i.e., desired robot explanations. A formal user study was needed to investigate this. After gaining knowledge of desired explanations, the robot designer must incorporate this knowledge to generate explanations. After explanations are generated, the robot must explicitly communicate them in the desired way by humans, along with modalities that are robot specific. For communication, robots should not only explain their current behaviors but also their past.

1.3 Approach

To investigate **desired robot explanations**, we drew inspiration from psychology, specifically human explanation studies. For a person who wants other people to explain, psychologists have found that people hope to get causal knowledge from explanations [105, 22] rather than from statistical evidence [72]. Koslowski [99] gives an example that both children and adults will not believe that the color of a car is a factor for gas mileage difference, rather only that the size of the car is. However, the belief is discarded when an explanation for how color affects the mood of a driver is given, which can change whether or not the car is driven in a fuel-efficient manner. With causal knowledge, understanding is improved, and “people can simulate counterfactual as well as future events under a variety of possible circumstances” [108].

Knowing that the purpose of seeking explanations is for causal knowledge, we then designed a user study by manipulating the amount of causal information provided in different execution conditions. I have designed and conducted the study on Amazon Mechanical Turk, the data is analyzed and results are presented in Chapter 3.

For **explanation generation**, a good starting point has been to investigate how a robot's internal states could be represented, from which we can find ways to convert them to external explanations. I started reviewing action sequence methods for robotic task specification and execution. I found the Behavior Tree approach is a better match because it is not only simple – using the analogy of trees and making the logic simpler to reason about and explain – but also powerful due to its expressiveness by having different node types to cover different use cases (See Chapter 2.3 and 4.2). Behavior Trees have also been presented to end-users for robot programming [133], utilized on Rethink Robotics Sawyer robots [40], and used in Learning from Demonstration tasks [65]. Paxton et al. [134] also showed that a system with BT was rated highly usable in a System Usability Scale (SUS) questionnaire, which paves the way to our explanation use case for end-users.

Nakawala et al. [122] discussed several other popular approaches for robot action sequence presentations, including ontology, state machines, and Peri Nets. Ontology belongs to the knowledge presentation family; this approach attempts to infer the task specification by high-level, abstract, underspecified input from a preset of actions, e.g., place the cup on the left of the plate. However, one drawback is that this knowledge presentation approach has less focus on how to specify tasks [97], leaving important specification details in a black box that is out of the developers' and users' control. State machines are overwhelmingly popular and extremely flexible, but this method has maintainability, scalability, and reusability issues due to its complexity [40], which are caused by interdependent states that are tightly coupled by transitions between states. Peri Nets are more of a technical underlying approach, specifically designed to tackle the concurrency issue by introducing operators and tokens, which in turn exposes unnecessary implementation details that humans would not explain.

Given the potential of Behavior Trees, I proposed a series of algorithms to frame Behavior Trees into a set of goals, subgoals, and steps to avoid deep hierarchy, and the shallow hierarchy is then used to generate high-level and failure explanations. This work is discussed in Chapter 4.

To **communicate explanations** to humans, robots should take advantage of robot-specific modalities for accurate communication. In the study of desired robot explanations, results show that participants thought nonverbal cues are confusing, including head and arm movement. Thus, methods for accurate externalization of robots' internal states are needed. In Chapter 2.5, I reviewed the related work for the use of display screens, augmented reality, and robot activities (i.e., non-verbal cues). We have found projection mapping has the potential for accurate externalization.

As opposed to the non-verbal methods and verbal explanations found among humans, projection mapping is a method that allows for direct and accurate externalization. This projection has the potential to completely remove the need for mental inference as the perceived objects or the objects to be manipulated are directly externalized. The directness is similar to the use of a display screen, but projection is more salient because bystanders or robot coworkers can also see the projection from farther away, instead of requiring people to stop their work in order to walk to a monitor to examine the robot's states. Direct projection onto the operating environment also has the potential to eliminate mental mapping from other media such as a monitor, which can cause misjudgment and lead to undesired consequences. We implemented projection mapping on a Fetch robot.

Finally, I conducted a human subjects study to look into how a robot could provide post-hoc explanations in long-term interaction scenarios to help people **infer missing causal information** due to environment change after a robot's past actions, including the projection communication method. This focus is interesting as the environment will highly likely change in long-term interaction scenarios and can lead to the loss of key causal information, e.g., a moved object after manipulation or a wet-floor sign as a ground obstacle removed after navigation.

To approach the problem, we again drew inspiration from psychology, especially the field of observational learning, which studies how to better learn attitudes, values, styles, and behavior by observing examples provided by others [17]. For robots, the acquisition of behavior is of particular interest and has been actively studied in the imitation subfield on how children and

adults imitate action sequences. Human imitation researchers [68] found that intentional actions with verbal markers (e.g., “there”) help aid causal inferences because adults and children assume the actions are made purposefully to reach a goal. Thus, I incorporated the verbal markers, and because projection markers can also be purposeful, projection is also included. Specifically, we investigated a combination of physical replay, verbal and projection indications, as well as each individually, resulting in 7 conditions. Chapter 6 presents the experiment design and the findings on whether participants were able to infer missing causal inference during the conditions.

1.4 Contributions

In summary, the primary contributions of this dissertation are shown below. The order follows the organization of this dissertation.

1. Through a user study and a strict replication study, we gain knowledge to better understand robot explanations that are preferred by humans. This step is important before we explore how to generate explanations. Major findings include
 - (a) Verbal explanations are needed to couple with non-verbal cues.
 - (b) Robots needs to get the attention of explainees before explaining.
 - (c) Explanation should be concise and happen when the robot failed.
2. Algorithms to semi-automatically generate explanations, including
 - (a) Modeling complex and simple robot tasks using simple yet powerful behavior trees (i.e., a complex mobile kitting task and a grid world task which is common for reinforcement learning).
 - (b) Framing behavior trees into four levels to generate shallow hierarchical explanations, as informed by the finding of concise explanation
 - (c) Making behavior trees dynamic to modify the robot’s task sequence after explanation

- (d) Failure explanation generation algorithms by leveraging behavior tree node types
3. Robot-agnostic implementation of projection mapping, a direct and instant method for a robot to communicate explanations.
- (a) Underlying principles from 3D point cloud input to 2D projection output
 - (b) Open-sourced code² with sample ROS nodes, Rviz configuration files, sample point cloud data, and ROS launch files
4. Through a human subjects study, we expand our knowledge of how a robot could aid people to infer past missing causal information in both manipulation and navigation tasks. This is important in a long-term human-robot interaction scenario where objects may be replaced or moved over time. Main findings are
- (a) Compared to physical replays, verbal and projection markers alone are almost always slightly worst.
 - (b) For picking, navigation, and placing location inferences, projection and verbal markers with arm movement should be used for correct inferences.
 - (c) To make picking inference faster, include physical replay.
 - (d) To accelerate inference of ground obstacle location in navigation, projection markers only are remarkably efficient.
 - (e) For placing inference, verbal indicators are exceptionally efficient.
 - (f) To increase people's confidence in their inferences, include physical replay in picking and navigation scenarios; For placement, include verbal indicators with replay.
 - (g) For lighter mental workload, use replay conditions.

²https://github.com/uml-robotics/projection_mapping

(h) To achieve better trust, include physical replay and projection; Verbal indicators are optional.

2 Related Work³

2.1 Desired Explanations

Before we evaluate our robot explanation system, we must investigate what do humans want robots to explain.

Psychologists have long studied human explanations. As categorized by Malle [108], humans either explain what was unexpected to have a complete and coherent understanding – meaning-finding explanations – or use explanations as communicative acts to create shared meaning and manage social interactions to influence the explainee’s mind or behavior – interaction-managing explanations. From an early age, people seek explanations for clarification and further understanding when they are surprised or confused [108, 90, 166]. In contrast to the unsettling experience due to not having any explanation from other people [71], Moerman [117] found that patients feel better when they receive explanations about their illness.

Researchers have also started to enable virtual agents to provide explanations. Ofra, Finale, and David [9] used the Pac-Man platform to summarize and explain Pac-Man’s turning behavior through videos. A human-subjects study [8] on this technique showed that participants preferred summaries from the model trained on participant-provided data rather than author-provided data. Indeed, artificial intelligence researchers (e.g., [113]) and the human-computer interaction community (e.g., [2]) are also contributing towards explainable or interpretable systems, including autonomous agents. Due to the embodiment of physical robots, some research in these fields (e.g., [11, 159]) is less applicable to human-robot interaction. In a literature review about explainable agents and robots [11], approximately half (47%) of the explanation systems examined, used text-based communication methods, which may be less relevant for robots that are not usually equipped with display screens. For some other work in the AI community (e.g., [116, 37]), however, the main audience has been machine learning experts interpreting trained models or the “black-box” systems

³Portions of this chapter appear in a paper [80] jointly authored with Jordan Allspaw, Adam Norton, and Dr. Holly Yanco. Please see Publication 1.

[4], although recently there is a shift to non-expert end-users for human-in-loop AI systems [172] and some in automated planning specifically [34].

However, not much is known about what robots should explain and how robots should produce verbal explanations as humans do. De Graaf and Malle [44] concisely discussed the theory behind human behavior explanation developed by psychologists in the past, and proposed to apply the theory behind human behavior explanation to autonomous intelligent systems. But robot embodiment and the difference between human and robot explanations remain largely unclear in the literature. Additionally, Thellman *et al.* [157] compared people’s interpretations of humanoid robot behavior to human behavior in static images with text descriptions and found that conscious goals are the perceived causes of robot behavior, while human behavior seems to be caused by dispositions. Hayes *et al.* [87] proposed explaining robot controller policy by manually annotating functions during programming, but this method is limited to programmers, and the explanations are constrained to programmers’ logic, which may not be helpful for non-expert or non-roboticist explainees. Chakraborti *et al.* [36] treated explanation as a problem to the suggestion of humans’ mental models of a robot ([153]) to align with the robot’s model. The proposed algorithm was claimed to generate explanations with desirable requirements, such as completeness, conciseness, monotonicity, and computability, which were mathematically formulated but not verified with human participants.

Instead of generating explanations directly and assuming certain qualities of explanations, we explore and attempt to better understand desired robot explanations by conducting a user study. Implicit nonverbal cues to express robot intent have also been investigated in the HRI community. Dragan *et al.* [56] proposed legible motion that adds extra robot arm motion to reveal a robot’s reaching intent to a specific object. Kwon *et al.* [100] used the similar concept of adding legible motion to indicate whether a robot could lift a cup or push a bookshelf. For some conditions, the added legible motion may have been unexpected to participants rather than helpful in aiding their understanding. While our focus has been on motion cues, for a comprehensive review of all

non-verbal cues, we refer readers to [32].

Like humans, the robot should be capable of explaining unexpected things. But, as Malle [108] pointed out, knowing why and what generates the behavior of a robot will further improve one's prediction of the resulting behavior, particularly when the cause is not easily apparent, such as in the handover task we designed for an almost-reachable cup. In addition, interaction-managing explanations must be verbally expressed [108], further suggesting implicit nonverbal cues (like motion cues) need to be accompanied by more explicit verbal explanations.

2.2 State Summarization

2.2.1 Manual Methods

A common approach is for developers to manually create categories by which the robot can explain its actions. For example, programmer specified function annotations for each designated robot action are used in [87]. By creating a set of robot actions, correlated with code functions, the system is able to snapshot the state of the robot before and after a function is called. Since the state of the robot could be exceedingly large in a real-world system, the state space is shrunk by isolating which variables are predetermined to be most relevant. These annotated variables are recorded every time a pre- and post-action snapshot is made. The robot then uses inspection to compare the pre- and post-variables of one action, compared to other similar successful actions, to make judgments.

Devin and Alami [50] described a Theory of Mind system to estimate the human partner's mental states but did not explore how the robot would express the assessment. In a shared pie cooking prep scenario task, Milliez et al. [115] proposed a simple tree-based system to embed explanation into parameterized task leaves for different human knowledge level (i.e., new, beginner and expert). As explaining the plan was not the focus, the work concentrated on determining whether to explain based on the human knowledge level. The robot would explain every step to

new collaborators, would ask beginners if they wanted an explanation, simply tell the current step to intermediate users, and offer no explanation or prompt at all for expert users.

Notably, there are quite a few tree-based methods. Kaptein et al. adapted hierarchical task analysis [143] to a goal hierarchy tree (GHT). This involves creating a tree where the top node would be a high-level task, which can be broken into a number of sub-goals, each linked by a belief (i.e., condition). Each sub-goal can then be broken into either sub-goals or actions. Choosing one sub-goal or action over another is based on a belief. The GHT can then be used to generate explanations. When comparing goal-based vs. belief-based explanations, Kaptein et al. found that adults significantly preferred goal-based explanations. However, this work focused on cognitive reasoning rather than behavior explanation or behavior programming and execution. While a NAO robot was used, the work did not involve any physicality such as arm or leg movement. In contrast, our work builds the foundation to provide a complete robot task specification and execution solution for robot developers who ultimately generate explanations either manually or by proposing state summarization algorithms.

2.2.2 Summarization Algorithms

While manually creating categories or explanations can be effective, it is time-consuming and not easily generalizable. Many techniques attempt to automate the process.

Programmer supplied explanations might be able to accurately describe the state of a robot, however, they can prove to be inadequate for a user. Ehsan et al. state that it is best to use a rationale justification [58] to explain to non-expert users, differentiating between a rationale and an explanation. An explanation can be made by exposing the inner workings of a system, but this type of explanation may not be understandable by non-experts. They suggest the alternative, a rationale, is meant to be an accessible and intuitive way of describing what the robot is doing. They also discuss how explanations can be tailored to optimize for different factors, including reliability, intelligibility, contextual accuracy, awareness and strategic detail; these factors can affect

the user's confidence, understandability of the explanations, and how human-like explanation was. The approach does not attempt to provide an explanation that reveals the underlying algorithm, but rather attempts to justify an action based on how a non-developer bystander would think. The authors explore two different explanation strategies. "Focused view rationale" provides concise and localized rationale and is more intelligible, and easier to understand. "Complete view rationale" provides detailed and holistic rationale and has better strategic detail and increased awareness.

Haidarian et al. proposed a metacognitive loop (MCL) architecture with a generalized metacognition module that monitors and controls the performance of the system [78]. Every decision performed by the system has a set of expectations and a set of corrections or corrective responses. Their framework does not attempt to monitor and respond to specific expectation failures which would require intricate knowledge of how the world works. However, the abandonment of intricate knowledge makes it difficult to provide specialized, highly detailed explanations to an expert operator.

Most of this prior work examined explanations within rule-based and logic-based AI systems, not addressing the quantitative nature of much of the AI used in HRI. More recent work on automatic explanations instead used Partially Observable Markov Decision Problems (POMDPs) which have seen success in several situations within robotics [162]. Unfortunately, the quantitative nature of these models and the complexity of their solution algorithms also make POMDP-reasoning opaque to people. Wang et al. proposed an approach to automatically generate natural-language explanations for POMDP-based reasoning, with predefined string representations of the potential actions, accompanied by the level of uncertainty, and the relative likelihood of outcomes. The system could also reveal information about its sensing abilities along with how accurate its sensor is likely to be. However, modeling using POMDPs can be time-consuming.

Miller discusses how explanations delivered to the user should be generated based on data from social and behavioral research, which could increase user understandability [114]. Whether the explanation is generated from expert developers or from a large dataset of novice operators,

both cases still require manually tying the robot algorithm to an explanation, a process that can be difficult and faulty.

In the literature review by Anjomshoae et al., they conclude that context-awareness and personalization remain under-researched despite having been determined to be key factors in explainable agency [11]. They also suggest that a multi-model explanation presentation is possibly useful, which would mean the underlying state representation would need to be robust enough to handle several different approaches. Finally, they propose that a robot should keep track of a user's knowledge, with the explanation generation model updated to reflect the evolution of user expertise.

As the more closed related work to Behavior Trees discussed below, we propose a set of algorithms to semi-automatically generate explanations from the Behavior Trees robotic task representations (see Chapter 4).

2.3 Robot Task Representation: Why Behavior Trees for Robot Explanations

For a robot to better perform tasks for which only humans are typically adept, robust task representations are important [122]. It is critical to have a readable and user-friendly representation for robot explanations, in order to minimize the mental burden of mapping robot task representation and execution to explanations. After perception, robot tasks can be decomposed to actions and action sequences for execution [77]. Nakawala *et al.* [122] listed different methods for robot action sequence presentations and discussed a few popular approaches including ontology, state machines, and Peri Nets. We will discuss the merits and drawbacks of each of these approaches and why behavior trees might be a better solution for robot programmers to manually provide robot explanations.

Ontology belongs to the knowledge representation family; this approach attempts to infer the task specification from high-level, abstract, underspecified input from a predefined set of actions, e.g., put the cup left of the plate. Two popular ontology implementations are KnowRob

[155, 19] and CRAM [20], which uses KnowRob. While CRAM⁴ uses the Common Lisp language, KnowRob embeds a number of common actions observed by leveraging the Prolog language and its logical predicates. As our previous work [80] discussed, KnowRob and CRAM introduce another programming paradigm, logical programming, to the robot system. This paradigm is distinctly different from C++ and Python used in the popular Robot Operating System (ROS) [137] (i.e., the procedural and object-oriented paradigms). Additionally, the knowledge representation approach does not examine how to specify tasks [134], leaving important details out of users' control.

State machines have a long history in computer science [142] and in robotics [128]. Unlike the traditional notion of a state, which is conventionally a world state, a state here is an execution in the task flow. A notable finite state machine (FSM) implementation is the SMACH ("State MACHine") library [23], which is tightly integrated with ROS. SMACH offers two interfaces: State and Container. A State represents an execution with a set of outcomes while a Container is a policy comprised of a set of states, which can be used to encapsulate behavior. SMACH allows the hierarchical composition of containers (i.e., state machines), which are also states with outcomes. States are transitioned through outcomes. Researchers have used SMACH to represent and execute tasks like serving drinks [24] and more general household tasks [124]. RAFCON [28] is another implementation of a hierarchical FSM, very similar to SMACH but with a well-designed graphical editor, including the ability to visualize deep hierarchies by panning and zooming. RAFCON has been demonstrated to specify complex tasks such as planetary exploration [27, 145].

While a state machine can describe taskset workflow and is very flexible and well-known, there are a few notable disadvantages. Colledanchise and Ögren [40] describe maintainability, scalability, and reusability issues. A workflow represented by a large number of states or hierarchies is hard to maintain, scale and is prone to design errors: adding a new state requires careful examinations [40] of incoming and outgoing transition dependencies, where the tight coupling of states makes it hard to reuse certain transition-states. For robotics specifically, it adds unnecessary

⁴<https://github.com/cram2/cram>

complexity especially for simple linear tasks without loops, such as a primitive pick manipulation task. In addition, a manipulation state machine breaks the fluency of such manipulation primitives, generating discrete arm movement that is paused, thus disconnected instead of a smooth multi-waypoint trajectory. Also as Nakawala *et al.* [122] points out, state machine implementations are code intensive and challenging for complex tasksets. For a comprehensive discussion of BTs and state machines, we refer readers to the Behavior Trees book by Colledanchise and Ögren [40].

Peri Nets (PNs) and Behavior Trees (BTs) are of growing interest. They share the same theoretical concepts of state machines and have equivalent state machine representations. Peri Nets, with sharing of tokens between states, are designed to be capable of modeling concurrency and distributed execution and coordination, with applications in multi-robot systems [173] and robot soccer [131]. However, concurrency can be achieved using framework-agnostic programming language constructs which are well-known across programming languages. Distributed execution can be achieved by using distributed frameworks like ROS or more general frameworks such as Apache Spark⁵ for big data. Nonetheless, our work does not focus on concurrency and distributed execution and coordination.

Compared to state machines, **Behavior Trees (BTs)** have some key advantages: modularity and reusability by behavior unit, expressiveness, and human readability by coherent and compact structure units [40]. Among state machines, Peri Nets and BTs, only the behavior tree approach is inherently user-friendly through human readability – the tree representation is simple, sharing familiar terminology of behaviors and using the analogy of a physical tree, removing the extra layer of abstraction of states and transitions, or operators and tokens. This allows for less information loss while we convert the underlying representation to explanations. In the last decade, BTs have been used for pick-and-place tasks [13] and end-user programming to instruct industrial tasks using a UR5 robot arm such as wire bending and sanding [76, 133]. Results of a user study on a BTs-based robot programming interface [134] indicate that BTs are a practical and effective

⁵<https://spark.apache.org/>

representation for specifying robot programs. Again for a comprehensive introduction, we refer readers to the Behavior Trees book published by Colledanchise and Ögren [40].

Thus, we choose BTs for hierarchical robot explanation generation, especially high-level tasks. Throughout this work, we also use BTs for primitive manipulation tasks after motion planning for simplicity. However, in Section 4.7, we also suggest using MoveIt task constructor (MTC) [73] for connecting sementicless waypoints in those manipulation primitive tasks, to not only achieve smooth multi-waypoint arm trajectory, but also to give us more information about the black-box probabilistic motion planning process, which provides potential for low-level explanations in the future.

2.4 Robotic Data Storage and Querying

Before generating explanations, a persistent storage system is needed to retain robot data to explain past events. The storage system must also have a query component for data retrieval, such as replay for accurate externalization when needed.

Terminal output or logs are common methods for debugging during active development and for error analysis after a robot has been deployed, but both methods have some drawbacks. Terminal output is essentially volatile memory, lost after the terminal window is closed, disallowing retrospection. However, despite being persistent on disks, software logs are unstructured and unlinked between related data, which makes it hard to effectively and efficiently query. Thus, researchers have been exploring database techniques to better store and query robotic data.

2.4.1 Storing Unprocessed Data

Many researchers have been leveraging the schemaless MongoDB database to store unprocessed data from sensors or communication messages from lower-level middleware such as motion planners [127, 21]. Being schemaless allows for recording different hierarchical data messages without declaring the hierarchy in the database (i.e., tables in relational databases such as MySQL). One

such hierarchical example is the popular Pose message type present in the Robot Operating System (ROS) framework [137]. A Pose message contains a position Point message and an orientation Quaternion message; a Point message contains float values x , y , and z ; an orientation message is represented by x , y , z , and w . It is imaginable to go through the cumbersome process of creating tables of Pose, Point, and Quaternion. Even more tables have to be created for each hierarchical data message. This advantage is also described as minimal configuration and allows evolving data structures to support innovation and development [127].

Niemueller et al. [127] open-sourced the *mongodb_log* library and are among the first to introduce MongoDB to robotics for logging purposes, which has applications to fault analysis and performance evaluation. In addition to being schemaless, the features that support scalability, such as capped collections, indexing and replication, are discussed. Capped collections handle limited storage capability by replacing old records with new ones. Indexing on a field or a combination of fields speeds up querying. Replication allows storing data across computers using the distributed pragma. Note that the indexing and replication features are also supported by relational databases.

While low-level data is needed, recording all raw data will soon hit the storage capacity limit: when old data is replaced by new records, the important information in the old data will be lost. This is particularly true when the data comes at a high rate; e.g., a HERB robot generates 0.1 GB per minute typically and 0.5 GB at peak times [127]. A more effective way is to be selective, only storing the data of interest [129]. However, storing raw sensor data only facilitates debugging purposes for developers; it does not solve the high-level explanation storage that will help non-expert users to understand the robot.

In addition, while it might be appropriate to expose the database to developers, a more effective way may be an interface that hides the database complexity, easing the cognitive burden on developers. This could be programming language agnostic, for example, by having an HTTP REST API or a ROS node. ROS is preferred as it is the most popular framework among roboticists and the communication layer has been implemented in various languages such as C++, Python,

C#⁶ and Go⁷.

Other researchers have also used MongoDB to store low-level data [20, 126, 169, 15] except for Oliveira et al. [129] who used LevelDB, a key-value database for perceived object data. Ravichandran et al. benchmarked major types of databases and found on average MongoDB has the best performance to continuous robotic data [139]. However, time-series and key-value databases are not included in the benchmark.

2.4.2 Storing Processed Data

Instead of looking for related data using the universal time range, Balint-Benczédi et al. proposed Common Analysis Structure to store linked data for manipulation tasks [15]. The structure includes timestamp, scene, image, and camera information. A scene has a viewpoint coordinate frame, annotations, and object hypotheses. Annotations are supporting planes or a semantic location, and object hypotheses are regions of raw data and their respective annotations. The authors considered storage space constraints, thus filtering and storing only regions of interest in unblurred images or point clouds. In their follow-up work [57], the Common Analysis Structure is used to optimize perception parameters by users providing ground truth labels.

Similarly, Oliveira et al. proposed a perception database using LevelDB to enable object category learning from users [129]. Instead of regions of raw point cloud data, user mediated key views of the same object are stored linking to one object category.

Wang et al. utilized a relational database as cloud robotics storage so multiple low-end robots can retrieve 3D laser scan data from a high-end robot, which has a laser sensor and its data being processed onboard with more storage and better computation power [161]. Specifically, low-end robots can send a query with their poses on a map to retrieve 3D map data and image data. PostgreSQL is used but the data structure detail is not discussed, as the paper focuses on resource

⁶<https://github.com/uml-robotics/ROS.NET>

⁷<http://wiki.ros.org/rosgo>

allocation and scheduling. However, a local data buffer on robots is proposed to store frequently accessed data to reduce the database access latency bottleneck.

Dietrich et al. used Cassandra to store and query 2D and 3D map data with spatial contexts such as building, floor, and room [52]. There are several benefits of using Cassandra, such as the ability to have a local server that can query both local data and remote data, avoiding single-point failure. Developers can also define TTLs (Time to Live) to remove data automatically, avoiding a maintenance burden.

In addition, Fourie et al. leveraged a graph database, Neo4j, to link vision sensor data stored in MongoDB to pose-keyed data [64]. Graph databases allow complex queries with spatial context for multiple mapping mobile robots, which enables multi-robot mapping. This line of research focuses on storing processed data but did not discuss a way to link raw data back to the processed data. This is important because not storing linked raw data may lead to loss of information during retrospection. There is a trend that other types of database systems, e.g., relational database (PostgreSQL) and key-value database (LevelDB and Cassandra) are used to store those processed data, because only a few ever-evolving data structures need to be stored.

2.4.3 Querying

There is no unified method for querying; most are application specific, such as efficient debugging [127, 15] and task representation [21, 156]. Interfaces are also tightly coupled to programming languages: JavaScript from MongoDB [127], Prolog [21, 156] and SQL [51].

In `mongodb_log`, Niemueller et al. proposed a knowledge hierarchy for manipulation tasks to enable efficient querying for debugging [127]. The knowledge hierarchy consists of all raw data and the poses of the robot and manipulated objects, all of which are timestamped. When a manipulation task fails, a top-down search is performed in the knowledge hierarchy in a specific time range. Poses are at the root of the hierarchy and raw data, such as coordinate frames and point cloud data, are replayed in a visualization tool for further investigation (i.e., Rviz in ROS). The

query language is JavaScript using the MapReduce paradigm, which supports the aggregation of data natively.

Beetz et al. proposed Open-EASE, a web interface for robotic knowledge representation and processing for developers [21, 156]. Robotics and AI researchers are able to encapsulate manipulation tasks semantically as temporal events with sets of predefined semantic predicates. Manipulation episodes are logged by storing low-level data, which are the environment model, object detection results and poses, and planned tasks in an XML-based Web Ontology Language (OWL) [158]. High-velocity raw data such as sensor data and robot poses are logged in a schema-less MongoDB database. Querying uses Prolog with a predefined concept vocabulary, similar to the semantic predicates.

While Open-EASE allows semantic querying, it does not come easily. One disadvantage is the introduction of a different programming paradigm, the logic programming in Prolog, which robot developers have to learn for querying regardless of the paradigm being used for robot programming. It is also unclear how to extend the predefined semantic predicates for other generic tasks in different environments.

Balint-Benczédi et al. use a similar high-level description language to replace the JavaScript query feature in MongoDB [15] to avoid the in-depth knowledge requirement of the internal data structure. The description language also contains predefined predicates and can be queried through Prolog. This work has the same drawbacks as Open-EASE.

Interestingly, Dietrich et al. proposed SelectScript, a SQL-inspired query-only language for robotic world models without having relevant tables in the database [51]. Without using a different programming language to specify how to retrieve data, SelectScript provides a declarative and language-agnostic way to specify what data are needed rather than how. SelectScript also features custom function support to queries and custom return type native to robotic applications such as an occupancy grid map.

While SelectScript is modeled on the well-known standard SQL, it is not language-agnostic

as stated. Custom functions are only supported in Python, leaving ROS C++ programmers behind. Except for requiring significant effort to support C++, it is not trivial to extend the return type to new data types such as the popular Octomap used in 3D mapping [93] for obstacle avoidance in SelectScript.

Fourie et al. proposed to use a graph database to query spatial data from multiple mobile robots [64]. However, it is not plausible for our use given that only one relationship is used: odometry poses linked to image and RGBD data. This work also suffers the same drawbacks of SelectScript in that custom queries have to be programmed in Java.

Similar to our argument in the previous storage sections, robotic database designers should embrace the programming languages with which robotic developers are already familiar. Database technology should be hidden by interfaces written in programming languages that also support access to the underlying database for advanced and customized use.

MongoDB's use has been proven by robotic developers and we thus choose to store low-level sensor data with it. This is mainly to replace the rosbag utility⁸ which relies on a filesystem and is not easy to query. Instead of writing from scratch, we used `mongodb_store`⁹, a ROS package of MongoDB interface to store raw and processed ROS messages to be replayed to MongoDB.

2.5 Human Interface for Communication

A human interface is used to communicate the explanations generated by the robot. Communication of the explanations can occur in different channels, such as a traditional graphical user interface (GUI) on a monitor, head-mounted displays, and robot movements. While some human interface methods have been studied for decades in the HCI community, especially GUI, the related work here is selective. We focus largely on novel approaches and the most prominent work, justified by the citation number relative to the publication year. There is a large body of research

⁸<http://wiki.ros.org/rosbag>

⁹https://wiki.ros.org/mongodb_store

for some techniques with existing comprehensive literature reviews, and we found the following papers are particularly useful: eye gaze in social robotics [6], using animation techniques with robots [144] (which provides 12 design guidelines), speech and natural language processing for robotics [112], and tactile communication via artificial skins in social robots [147].

2.5.1 Display Screen

While computer interfaces largely use a display screen for the primary communication channel, screens on robots are largely used to display facial expressions [95] due to their physicality, but are considered less convenient than speech [45]. For co-location scenarios, it is rare to find a display screen as part of a robot that is not attached to its head, so very little research has been performed for simple displays or visualizations of sensor values or other relevant information.

Brooks investigated displaying a general set of state icons on the body of robots to indicate internal states [25]. Five icons – OK, Help, Off, Safe, and Dangerous – were shown to participants for evaluation. The results show that while bystanders are able to understand those icons, their level of understanding is vague. For example, the “Off” icon could be interpreted as stating that the robot is powered off or that it is just not currently actively operating.

SoftBank Robotics’ Pepper robot is one of the few robot systems that features a touch screen not attached to its head. Feingold-Polak et al. found that people enjoyed interacting with a touch screen on a robot more than using a computer screen with a keyboard [60]. Specifically, participants preferred to use the touch screen to indicate the completion of a task. de Jong et al. used Pepper’s screen to present buttons to use for inputting instructions, such as object directions [45]. Bruno et al. used Pepper’s touch screen like a tablet where multiple-choice questions are shown and users can answer by tapping on the choices [29].

While a display screen has been demonstrated to be effective at showing accurate information (e.g., replaying past events [94], which can be used during explanations), there is sometimes a mental conversion issue where humans have to map what is displayed on the screen to the physical

environment. A display screen may also suffer from being less readable from a longer distance, which is important as such proximity to a robot may not be safe during certain failure cases [91].

2.5.2 Augmented Reality (AR)

Utilizing AR for explainability allows visual cues to be projected directly into the environment with which the robot interacts, allowing for more specificity and reference points to be drawn. This technique can make explanations more accurate, less ambiguous, and remove the burden of mental mapping between different reference frames (e.g., 2D display screen compared to the real-world 3D environment).

Andersen et al. proposed to use a projector to communicate a robot's intent and task information onto the workspace to facilitate human-robot collaboration in a manufacturing environment [10]. The robot locates a physical car door using an edge-based detection method, then projects visualizations of parts onto it to indicate its perception and intended manipulation actions; before part manipulation, the robot will project the segment of interest to inform the worker of its next manipulation target. The authors also conducted an experiment by asking participants to collaboratively rotate and move cubes with the robot arm, comparing the AR projector method to the use of a display screen with text. Results show there were fewer performance errors and questions asked by the participants when using the projector method.

For mobility, researchers also leveraged projection techniques onto the ground to show robot intention. Chadalavada et al. projected a green line to indicate the planned path and two white lines to the left and right of the robot to visualize the collision avoidance range of the robot [33]. Gradient light bands have also been used to show a robot's path [165]. Similarly, Coover et al. projected arrows to show the robot's path [41], while Daily et al. used a head-mounted display to visualize the robot's path onto the user's view of the environment [42]. Circles have also been used to show landmarks on a robot swarm using a projector located above the performance space [69]. However, the AR techniques utilized in these papers are passive and not interactive. Chakraborti

et al. proposed using Microsoft HoloLens to enable a user to interact with AR projections [35], where users can use pinch gestures to move a robot’s arm or base, start or stop robot movement, and pick a block for stacking. Gao and Huang [67] proposed using a projector to project an interface on a flat tabletop surface for robot programming. However, the focus is on the ease of robot programming, and the projection does not require mapping objects back to their corresponding real-world objects.

2.5.3 Robot Activities

Due to the physicality of robot systems, body language of robots has been studied extensively in the HRI community to communicate intent. For example, Dragan et al. proposed using legible robot arm movements to allow people to quickly infer the robot’s next grasp target [55]. Repeated arm movement has also been proposed to communicate a robot’s incapability to pick up an object [100]. Eye gaze behavior or head movement has also been studied (e.g., [118, 6]). However, this communication method is limited in the amount of information that it can convey, if used as the only channel of communication.

In addition to robot movements, researchers have also explored auxiliary methods of communication such as light. Notably, the Rethink Robotics Baxter system utilizes a ring of lights on its head to indicate the distance of humans moving nearby to support safe HRI. Similarly, Szafir et al. used light to indicate the flying direction of a drone when co-located with humans in close proximity [152]; the results show improvement in response time and accuracy.

In our work, we used projection mapping, which belongs to AR. As opposed to the non-verbal methods and verbal explanations found among humans, projection mapping is a method that allows for *direct and accurate* externalization. This projection completely removes the need for mental inference as the perceived objects or the objects to be manipulated are directly externalized. The directness is similar to the use of a display screen, but projection is more *salient* because bystanders or robot coworkers can also see the projection from farther away, instead of requiring

people to stop their work in order to walk to a monitor to examine the robot's states. Direct projection onto the operating environment also *eliminates mental mapping* from other media such as a monitor, which can cause misjudgment and lead to undesired consequences, resulted from the robot's actions to the world.

We have finished implementing projection mapping (Chapter 5), paving the way for the replay experiment to be discussed in Chapter 6.

2.6 Observational Learning

To assist our investigation on how a robot can help people to infer past missing causal information, we looked for inspiration from the observational learning field from Psychology. The field focuses on human cognitive evaluations of observed actions of humans, as opposed to robots studied in human-robot interaction (HRI). Observational learning itself is defined as being “concerned with the acquisition of attitudes, values, and styles of thinking and behaving through observation of the examples provided by others” [17]. Applied to robots that communicate explanations with humans, we can get insight into how action sequences robots should potentially communicate to others for them to understand the most important information the robot conveys, to leave a greater impression.

One subfield in observation learning studies how children imitate action sequences, i.e., behaving, from others [30]. There are two schools of thought: (1) children are rational learners (rational imitation [123]) who reproduce or imitate the most salient causal actions to an outcome; (2) children tend to overimitate others [106, 88], copying unnecessary, non-causal yet normative behavior in a social and cultural setting [97].

For over-imitation, it has been validated that children and adults both over-imitate a stranger's actions even when they are not aware of being participated in an experiment [167]. It is worth noting that the experiment was conducted in a real-world setting to avoid the adult participants being sensitive to a laboratory environment [31].

The first school of thought, the rational-imitation paradigm, has also been studied with human adult participants. Buttelmann et al. [31] replicated the head-touch demonstration experiment with eye-tracking equipment, where a *model* head-touch lamp when hands are occupied versus hand-touch when hands are free. The results show that there are no significant differences between adults and infants in attention, the amount of looking at the modeled action. The only significant difference is that adults look at the model's head in the video demonstrations significantly longer than infants (around 1 minute and 20 seconds), while no difference is found for looking at the torso (which takes only a few seconds).

In our work, we are particularly interested in the first paradigm, rational-imitation, partly because a recent study [148] found that children overimitate robots less than humans due to the lack of social motivations.

On the model side, researchers show the demonstrator's intentional and knowledge state will be used to aid the causal inferences [68]. In [68], results suggested that intentional actions with verbal markers (e.g., "There") are assumed by children that the model act purposefully to reach a goal. Thus the intentional actions help to understand causality. Inspired by this, we incorporated verbal markers into some of our conditions.

Regarding video studies, televised models showing an action sequence are not rare and have been proven to be effective at an early age. In a study [121], 10- and 12-month-old infants are able to learn and imitate negative and positive emotional reactions from the televised model. Another study showed that 12-month-olds were also able to perform rational head-touch imitation during constrained conditions (i.e., hands-occupied), compared to hand-touch, also from a televised model [174]. During approximately the same year, researchers found that 2-year-old toddlers could learn verbal labeling during an action sequence with repetition from television after a 24-hour delay, and both video parent labeling and video voice-over labeling did not differ from live parent labeling [18]. In another televised experiment, results show similar results that toddlers can use acoustical action effect from both a live mode and televised model [98]. This line of work suggests that the

action sequence in a video accompanied by the acoustical action effect should be on par with a live demonstration. Inspired by the acoustical effect, we become creative and also explore the projection mapping method in addition to verbal markers, because projection mapping is not a method that humans have and can be seen as a salient effect.

3 Desired Robot Explanation¹⁰

3.1 Introduction

Explaining our behavior is a natural part of daily life and the lack of explanations can be unsettling and disturbing [71]. It is important to provide explanations to improve understanding. This need is especially true for robots, often designed to be perceived as intelligent entities with physical appearances that resemble humans. Past work in Human-Robot Interaction (HRI) has illustrated that improving understanding of a robot makes it more trustworthy [46] and the resulting interaction more efficient [7]. Human-agent interaction researchers in the artificial intelligence community have started to enable virtual agents to provide explanations [9]. Unlike virtual agents, however, robots have physical embodiment, which has been shown to have effects on metrics such as empathy [146] and cooperation [14].

Yet to our best knowledge, it remains unclear how robots should be enabled to *verbally* explain intention and behavior, or how to couple verbal explanations with other embodied explanatory cues, despite the growing collection of HRI research on non-verbal motion cues to indicate a robot's intent (e.g., [56, 118, 100]). Before empowering robots to explain themselves, we need to answer *what and how would humans like a robot to explain?* In this study, we investigated participants' desired verbal explanations after being presented with robot non-verbal motion cues.

We drew inspiration from psychology: Specifically, research that examines the purposes of desired explanations from another. Psychologists have found that people hope to gain causal knowledge from explanations [105, 22] rather than from statistical evidence [72]. Koslowski [99] illustrated that when told that a car's color was a determining factor in the car's gas mileage, children and adults did not believe this to be true. Rather, they only believed that the size of the car impacted gas mileage. Their beliefs could be changed, however, when given the explanation that car color affects the mood of the driver which can change whether or not the car is driven

¹⁰This chapter appears in a workshop paper [85] jointly authored with Dr. Holly Yanco and a journal paper [81] jointly authored with Dr. Elizabeth Phillips and Dr. Holly Yanco. Please see Publication 2 and 3.

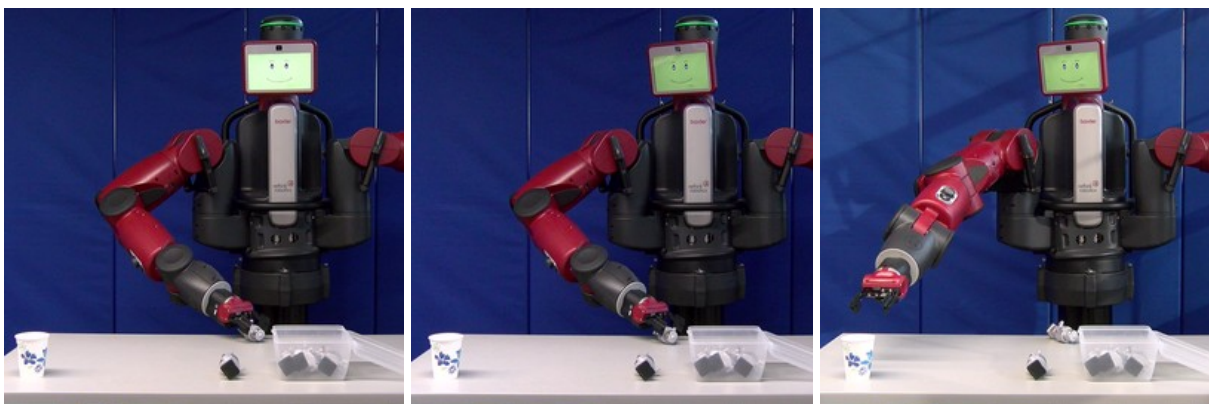


Figure 1: Three of the six handover conditions after the almost-reachable cup is detected, without the headshake included. *Left*: Robot does nothing (No Cue). *Middle*: Robot’s head turns towards the cup (Look). *Right*: Robot’s head turns towards the cup and its right arm is extended repeatedly (Look & Point).

in a fuel-efficient manner. With causal knowledge, understanding is improved, and “people can simulate counterfactual as well as future events under a variety of possible circumstances” [108].

Knowing that the purpose of seeking explanations is often for causal knowledge, we designed and conducted an online experiment using Amazon’s Mechanical Turk (MTurk) to investigate the perceived need for robot explanations, whether that perceived need is similar to what we would expect of a human, as well as how people want a robot to provide such explanations even when non-verbal causal motion cues are provided by the robot.

Participants first watched a scenario in which a Baxter robot was not able to hand over a cup. Information about why the robot was not able to complete the cup handover (causal information) was missing, i.e., the fact that the cup was slightly out of reach was not known to participants.

Participants then watched one of six reactions (Figure 1) from the robot containing motion cues intended to convey why the robot was unable to complete the handover task: doing nothing (No Cue), turning its head to the cup (Look), or turning its head to the cup with the addition of repeated arm movement pointed towards the cup (Look & Point) and each of these with or without the addition of the robot providing a Headshake.

In a questionnaire, we asked participants if they perceived unexpected things throughout

the task that they felt the robot should verbally explain. We then asked about some of the desired properties and the content of the robot explanations following a failed handover. Specifically, we asked about desired explanation timing (at the end, in situ, a priori, or other), engagement importance, whether and how robot explanations should be different from human explanations, explanation summarization, and explanation verbosity.

Results revealed that participants felt robot behavior should be explained in all conditions. The addition of a headshake and arm movement motion cues to indicate that the robot could not complete the task did not result in reduced need for explanations or less perceived unexpectedness, but rather confused participants, as the headshake was often interpreted as the robot having disobeyed the handover request, and the intention of the arm movement was unclear to participants.

Regarding the properties of robot explanations, participants reported that they wanted robots to explain in situ, not at the end of the task. They also thought that engagement was important. The robot should get the participants' attention by looking at them and possibly address them by name before explaining, which needs more investigation. People thought robots should explain the same content as humans explain, wanted concise summaries, and were willing to ask a few (1 to 3) follow-up questions after a summarized explanation was provided by the robot in order to gain more information. Participants thought the robot should explain why it failed to complete the task, why it disobeyed them during headshake executions without any arm motion, and why it kept moving its arm. When the robot did nothing, people wanted to know about the robot's previous behavior.

3.2 Hypotheses

Driven by the prior related work, partly on the motion cues, we formalized the following hypotheses.

3.2.1 The Need For Robot Explanations

Hypothesis 3.1 – Robot behavior needs to be explained. In general, robot behavior will be considered unexpected to people, and there will be a desire for it to be explained.

Hypothesis 3.2 – As more causal information about robot behavior is provided, there will be less need for an explanation from the robot. There will be an association between the amount of causal information provided in robot behavior and participants' reported need for explanation. Specifically, as the robot's behavior provides participants with more causal information, participants will report less need for explanation.

Hypothesis 3.3 – Adding a headshake to the robot's explanation will result in less need for an explanation. Including a negative headshake, which implies that the robot cannot complete the handover task, will give participants more information about the robot's behavior than execution alone. When the robot couples its behavior with a headshake, participants will report less need for explanation than when it does not.

3.2.2 Expected Properties of Robot Explanations

Hypothesis 3.4 – Explanations offered at multiple points in time will be desirable. Having robots explain a priori, in situ, and at the end will be more desirable than at any one single point in time.

Hypothesis 3.5 – Engagement prior to providing an explanation will be important. People will prefer that the robot get their attention prior to explaining behavior as opposed to explaining behavior without getting their attention.

Hypothesis 3.6 – Similarity to human explanations will be expected. Participants will report that they expect that robot explanations should be similar to human explanations

Hypothesis 3.7 – Summarization will be preferred. Participants will prefer that the robot pro-

vides an explanation that is presented as a summary as opposed to a detailed explanation.

Hypothesis 3.8 – Fewer number of follow-ups will be preferred. Participants will prefer to ask fewer clarifying follow-up questions as opposed to more follow-up questions after an explanation is given from the robot.

3.3 Method

3.3.1 Power Analysis, Participants, and Participant Recruitment

We used G*Power 3.1.9.4 [59] to perform two *a priori* power analyses because we planned to run two types of hypothesis tests.

We first performed an *a priori* power analysis for “Goodness-of-fit tests: Contingency tables”. The parameters were: Effect size $w = 0.5$ for large effect size, α error probability = 0.05, Power ($1 - \beta$ error probability) = 0.95, Df = 2 which reflected the number of fixed choices in our measures described in Section 3.3.3. The output parameters in G*Power showed that the sample size to reach desired power $1 - \beta = 0.95$ was 62 for a single goodness-of-fit test. Thus, for the 6 conditions of our 3×2 experiment design, we needed at least $6 \times 62 = 372$ participants.

We also performed an *a priori* power analysis for “ANOVA: fixed effects, special, main effects and interactions” tests. The parameters were: Effect size $f = 0.4$ for large effect size, α error probability = 0.05, Power ($1 - \beta$ error probability) = 0.95, Numerator df = 2 (maximum for main effects and interactions), and Number of groups = 6, reflecting the number of independent conditions in our study. The output parameters showed that the total sample size needed was 100.

Thus our study would need approximately $N = 372$ participants to be sufficiently powered for both types of statistical tests.

Using Amazon Mechanical Turk (MTurk), we recruited a total of 460 MTurk workers to participate in the study. We purposefully recruited extra participants to account for potential data loss due to some MTurk workers failing data quality assurance checks and/or not completing the

entire study. All of the 460 participants completed the study. However, we had two participants complete the study twice, providing us with 458 unique cases. Seventy-eight participants did not pass the data quality assurance checks in the form of “attention check” questions (described below), which resulted in 380 valid cases used in data analyses. To ensure that we had an equal number of participants in each of the 6 conditions, we trimmed the data from the last participants who entered into the study. This procedure resulted in a sample size of $N = 372$ with 62 participants in each of the 6 between-subjects conditions.

However, after completing a strict replication study (see Section 3.5) and re-inspecting our data, we found 4 participants who responded to a question by providing a partial or full copy and paste of the common definition of a robot giving when searching the web for the word “robot” (e.g., entry for robot on Wikipedia). We thus removed these four participants and then tried to replace their data with data from four participants who had previously been trimmed from the original dataset, as described above. However, we were unable to replace data equally across conditions. Therefore, we trimmed the data again to balance the number of participants in each of the experimental conditions. This process resulted in 61 participants in each condition, $N = 366$. The analyses reported in subsequent sections were conducted on this final dataset.

The final sample included 209 males, 153 females, 3 participants who preferred not to say, and 1 transgender person; with ages ranging from 18–74, $M = 37$, $median = 34$, $skewness = 1.07$. Seventy-three participants (20%) agreed with the statement, “I have experience with robots,” 204 disagreed (56%), and 89 (24%) responded that they neither agreed nor disagreed.

Specified qualifications for participation on MTurk included being over 18 years old, living in the United States, which provided a reasonable assumption of some English language comprehension, having performed at least 1000 Human Intelligence Tasks (HITs), and a 95% approval rating. Each MTurk worker, whether or not they passed data quality assurance checks, was paid U.S. \$1 for their participation.

Table 1: Explanation Measure Items

| |
|---|
| <p>Unexpectedness (Cronbach's $\alpha = 0.80$)</p> <ol style="list-style-type: none"> 1. <i>I found the robot's behavior confusing.</i> 2. <i>The robot's behavior matched what I expected.</i> (Reversed) 3. <i>The robot's behavior surprised me.</i> |
| <p>Need (Cronbach's $\alpha = 0.74$)</p> <ol style="list-style-type: none"> 1. <i>I want the robot to explain its behavior.</i> 2. <i>The robot should not explain anything about its behavior.</i> (Reversed) |
| <p>Human-Robot Difference (Cronbach's $\alpha = 0.49$)</p> <ol style="list-style-type: none"> 1. <i>There should be no difference between what a robot says to explain its behavior and what a person would say to explain the same behavior.</i> 2. <i>If a person did what the robot did, they should both explain the same behavior in the same way.</i> |
| <p>Summarization (Cronbach's $\alpha = -0.57$; 0.65 if Q1 is dropped)</p> <ol style="list-style-type: none"> 1. <i>The robot should give a very detailed explanation.</i> (Reversed) 2. <i>The robot should concisely explain its behavior.</i> 3. <i>The robot should give a summary about its behavior before giving more detail.</i> |
| <p>* Likert items are coded as -3 (Strongly Disagree), -2 (Disagree), -1 (Moderately Disagree), 0 (Neutral), 1 (Moderately Agree), 2 (Agree), and 3 (Strongly Agree).</p> |

3.3.2 Robot Platform

A Rethink Robotics Baxter humanoid robot (humanlikeness score = 27.30 on a scale of 0 “Not human-like at all” to 100 “Just like a human” [136, 62]) depicting a digital smiling face from [61, 84] was used in the experiment. Baxter has a large appearance: 1.8m tall with two 128cm arms, measured from its shoulder joint to gripper tip. It is taller than an average adult male over age 20 in the U.S. (175 cm) [66], and Baxter's arm is around two times longer than an average human arm [104].

3.3.3 Measures

To test our hypotheses, we designed a measure consisting of items about participant perceptions of robot explanations. To create the measure, we searched the existing HRI and robotics literature for scales in the context of robot explanation, but few have been developed and validated by the com-

munity. However, we did find a subjective scale of predictability in [54] by Dragan and Srinivasa; because of its high internal reliability (Chronbach's $\alpha = 0.91$), we adapted two of its questions (Table 1 in [54]) for our Unexpectedness measure (the last 2 questions in the first row in Table 1):

1. "The robot's behavior matched what I expected" is adapted from "Trajectory 'x' matched what I expected".
2. "The robot's behavior *surprised* me" is adapted from "I would be *surprised* if the robot executed Trajectory 'x' in this situation".

Our measure consisted of four subscales, each with items tailored to gather information about participant perceptions of the unexpectedness of the robot's behavior, the need for explanations, desired similarities and differences between robot and human explanations, and the desired level of detail of robot explanations. For each subscale, multiple contradicting or very similar questions were designed to help establish internal consistency. All items are listed in Table 1. Participants responded to each item using a 7-point Likert-type scale that ranged from -3 "Strongly Disagree" to 3 "Strongly Agree," with 0 representing neither agreement nor disagreement at the mid-point of the scale. The order of these items was presented to participants at random.

Additionally, we asked participants to respond to several items about desired properties of robot explanations. These included questions about explanation timing, engagement, and summarization, as follows:

- *Timing* (forced-choice). When would be the best time for the robot to explain its behavior? Participants responded with "At the end"; "Whenever something unexpected happens"; "Before something unexpected happens"; or "Other (Please elaborate)."
- *Engagement Importance* (true/false). Do you think it is important for the robot to get your attention before starting to explain its behavior?

- *Engagement Approach* (forced-choice). How should the robot get your attention before starting to explain its behavior? Possible responses included, “Look at me”; “Raise volume”; or “Other (Please elaborate).”
- *Summarization* (forced-choice). After the robot gives a summary about its behavior, how many questions would you be willing to ask to get more details from the robot? Response choices included, “None”; “A few (1 to 3)”; “As many as needed (4 or more).”
- *Content* (open-ended). What would you like the robot to explain specifically?
- *Content Reasoning* (open-ended). Please explain why. (Why would you like the robot to explain the things you mentioned above?)

Finally, we asked several attention check questions to help us ensure participant attention to the experimental stimuli, similar to those used in Brooks *et al.* [26]. Participants were asked to indicate the color of the robot depicted in the experimental stimuli (Red and black; Blue and yellow; Green and white), the identity of certain objects in the scene: if there were any robots shown in the scene (Yes; No), and if the robot moved (Yes; No). Failure to answer any of those questions correctly resulted in the removal of the participant’s data from the analysis. These attention check questions were randomly intermixed with the items asking about robot explanations and the properties of robot explanations.

3.3.4 Experimental Design

This study followed a 3×2 between-subjects design. Participants were asked to imagine that they were interacting with the Baxter robot while viewing a video of Baxter executing a handover task. In the handover task, the robot attempted to grasp and hand over a cup under one of the study conditions. The handover task was chosen because robots will often be expected to complete handover tasks, especially as they enter factories and homes. The handover task is one of the

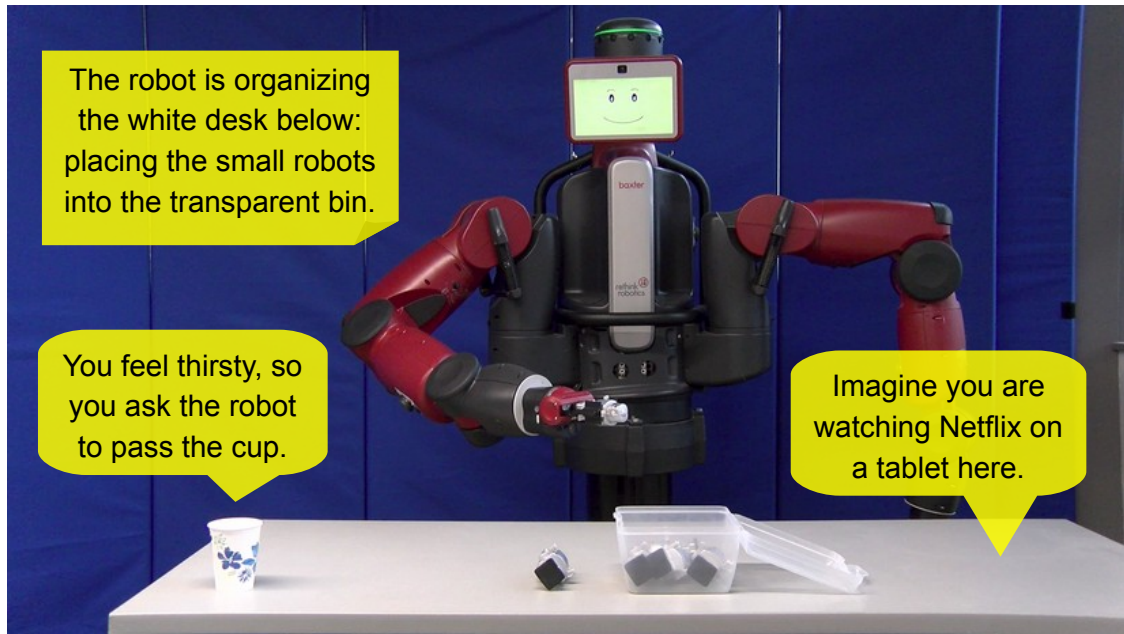


Figure 2: To establish the handover task context, the image above was first presented to participants with the three yellow pop-ups slowly fading in one after another clockwise from the top left.

most common interactions among humans. Additionally, the handover task involves manipulation, which is characteristic of typical tasks that many robots currently do. We manipulated the robot's handover task *execution type*. The execution type conditions were designed to provide participants with a variety of robot motion cues giving causal information for why the robot was unable to complete the handover task in each video. We also manipulated whether or not the robot shook its head negatively (*Headshake*) while completing each execution type. This resulted in 6 between-subjects experimental conditions. The code for this study is available on GitHub¹¹.

Study Conditions For each condition, participants were provided with a video (see Figure 2), where they were informed that the Baxter robot was organizing a desk by placing several small toy robots into a transparent bin. Participants were then told to imagine that they were with Baxter in the scene by imagining that they were busy watching Netflix near Baxter. They were then asked to imagine that they were thirsty and have asked Baxter to pass them a cup.

¹¹<https://github.com/uml-robotics/takeit>

Table 2: Study Conditions Across The Two Factors

| Execution Type: <i>Motion Cues</i> | Headshake |
|---|--------------------------|
| Look & Point: <i>Head turning & arm movement</i> | <i>With Headshake</i> |
| Look: <i>Head turning</i> | <i>Without Headshake</i> |
| No Cue: <i>None of above</i> | |

Also seen in Figure 2, in the video, the three pieces of information described above were presented by pop-up text box annotations: From the top left, each pop-up text box slowly faded in, in clockwise order, and displayed for 5 seconds. The duration was selected to make sure that participants had ample time to finish reading a pop-up before the next was shown. Participants could also pause the video to review the information.

Note that for the pop-up content, only the left-bottom pop-up (ask the robot to pass the cup) was important to establish the handover task. The other two were intended to make the imagined interaction in the scenario more complete and used to remove potential bots or careless responders via attention check questions.

After viewing these annotations, the participants viewed the following white text on a black background for 6 seconds: “The robot understands your request. Now please watch how the robot responds.” Note that we carefully selected the neutral word of “understand” rather than “acknowledge” or “accept” to avoid the impression that the robot would unquestionably finish the handover task. After presenting this information, the video depicted the Baxter dropping the small toy robot it was previously holding and about to place in the transparent bin. Then Baxter attempted to pass the cup under one of the study conditions described below.

Table 2 briefly lists the experimental conditions across the two factors *Execution type* and *Headshake*. The videos shown to participants are available on YouTube¹². A brief summary of each of the study conditions is provided below.

- *Look & Point without Headshake*. Inspired by the legible arm motion research in [56, 100],

¹²https://www.youtube.com/playlist?list=PLnvaJwyK3MF-x0at0pBEq_PhbheI4fMS7

the Look & Point execution conditions were designed to provide the most causal information about why the robot was not able to complete the handover task. The robot provided motion cues to help participants infer causal information for why the robot could not execute the cup handover task. Specifically, in this condition, during the handover task, the robot would *stop organizing, move its arm to reach towards the cup, simultaneously move its head to look at the cup, and keep extending its arm fully toward the cup*. However, it could not reach the cup on the table. The robot repeated this pattern of motion cues towards the cup three times, which was intended to show participants that the robot was trying to complete the task but was unable to do so because the cup was out of reach. Note that this type of motion is not what a robot would commonly do when it cannot complete a grasping or handover task; commonly the robot would simply stop organizing and do nothing. Typically its motion planner, e.g., MoveIt [38], would return a plan or execution failure status when the object is not reachable, rather than physically illustrating that an object is out of reach.

- *Look & Point with Headshake*. Under this condition, we added a Headshake to the Look & Point motion execution. In addition to arm and head motion cues described above, after reaching for the cup, the robot shook its head from left to right (i.e., a “No” pattern) to further communicate that it could not reach the cup. The robot repeated the reaching motion while shaking its head two more times before the video ended.
- *Look without Headshake*. In the Look execution condition, during the handover task, the robot would *stop organizing, and turn its head towards the cup*. The robot did not reach its arm toward the cup. Roboticists may immediately understand why the robot was unable to complete the handover task: The robot turns its camera mounted on its head, probably an RGBD camera, to detect the cup with depth information, plans to maneuver its arm to the pose of the cup, and eventually failed to do so because the cup was out of reach. However, because roboticists represent a small group of specialists, this execution type is likely opaque

to most people, who will be wondering why the robot turned its head toward the cup but did not pick it up.

- *Look with Headshake.* Similar to the Look & Point with Headshake condition, in the Look with Headshake condition we added a Headshake to the robot’s looking motion cue. Specifically, the robot would *stop organizing, turn its head towards the cup, and shake its head negatively.* The robot did not reach its arm toward the cup.
- *No Cue without Headshake.* Under this condition, during the handover task, the robot would *stop organizing and do nothing.* Unlike the previous two execution conditions, the robot did not provide any motion cues, e.g., turn its head or reach with its arm.
- *No Cue with Headshake.* Again we added a Headshake to the robot’s execution. In the handover task, rather than doing nothing, the robot would *stop organizing and shake its head from left to right.* Note that, unlike the Look with Headshake condition, the robot did not turn its head toward the cup before shaking its head in a “No” pattern.

3.3.5 Procedure

The study was conducted on Amazon’s Mechanical Turk (MTurk) where participants entered the study via an anonymous link to a Qualtrics survey. Once started, participants were presented with informed consent information. After reviewing this information and agreeing to participate, participants were randomly assigned to one of the experimental conditions. On Qualtrics, participants were provided with the following instruction, “Before answering questions, please watch the following video in full-screen mode.” Participants then watched one video depicting the robot executing the handover task under one of the experimental conditions. After viewing the entire video, participants were then asked to complete the measure containing the explanation, properties of explanations, and attention check items. On the survey page, participants were encouraged to review the video as many times as they needed. Specifically, they were told: “You may go back

to the video by clicking the back button at the bottom of this page. If you need to do this, your answers on this page will be saved.” Participants were then provided with a code to receive their payment. The entire study took approximately 10 minutes to complete for most participants, who were compensated U.S. \$1 in return for participating in the study. This study was approved by the Institutional Review Board at the University of Massachusetts Lowell.

3.4 Results

We used R to analyze the data. Table 1 lists all of the items from the robot explanations measure, the anchors for the Likert-type scales, and Cronbach’s alpha values of internal consistency for each of the items. M used without standard deviation values denotes median values throughout this section.

3.4.1 H3.1: Robot behavior will be considered unexpected and needs to be explained

Cronbach’s alpha on the unexpectedness subscale was 0.80, a good level of internal consistency reliability [49]. We thus calculated an unweighted average score of responses across the items to achieve a composite score for the unexpectedness scale on our questionnaire, plotted in Figure 3 and 4.

To analyze the data, we used a two-way between-subjects factorial ANOVA with the unexpectedness score as a dependent variable.

We did not find a statistically significant interaction between *Headshake* and *Execution Type*, but found statistically significant main effects for *Headshake* ($F(1, 360) = 15.91, p < 0.0001$) and *Execution Type* ($F(2, 360) = 11.90, p < 0.0001$) on unexpectedness scores.

Before we conducted pairwise comparisons across conditions, we used the ANOVA model to calculate estimated marginal means of all conditions and performed multiple comparisons with Holm-Bonferroni correction [89] to test whether these means significantly deviate from 0 ($H_0 : \mu = 0$). Without Headshake, we only found a statistically significant difference in No Cue ex-

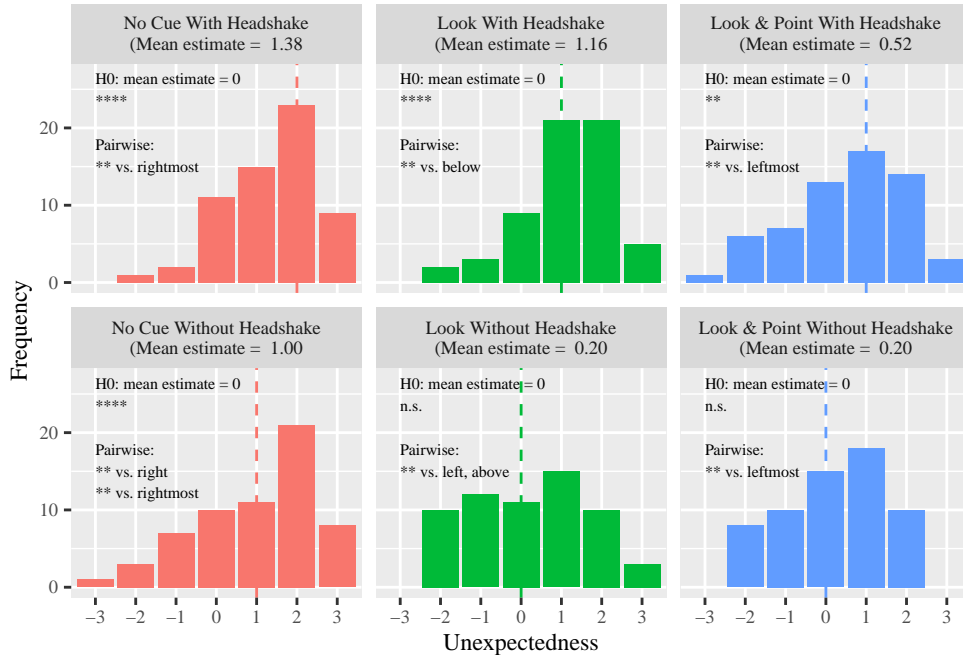


Figure 3: The distribution of Unexpectedness responses with median lines and estimated marginal means. Except for Look without Headshake and Look & Point without Headshake, all conditions were rated unexpected (first two lines of the annotation in each of the boxes, comparison against 0). Results of post-hoc pairwise comparisons are also shown (second and third lines of the annotation in each of the boxes).

ecution type ($1.00 \pm 0.17, p < 0.0001$), but not in Look ($0.20 \pm 0.17, n.s.$) and Look & Point ($0.20 \pm 0.17, n.s.$) conditions. With Headshake, statistically significant differences were found in all execution types – No Cue ($1.38 \pm 0.17, p < 0.0001$), Look ($1.16 \pm 0.17, p < 0.0001$) and Look & Point ($0.52 \pm 0.17, p < 0.01$).

The results above suggest that, when the robot performed the Look & Point and Look executions without Headshake, participants felt neutral about the unexpectedness (neither unexpected nor expected), but participants reported that the robot’s behavior was significantly more unexpected for the No Cue execution than the other two execution types. Surprisingly, the addition of the Headshake behavior did not decrease the unexpectedness but rather made the robot’s behavior more unexpected with strong evidence in the No Cue and Look with Headshake conditions and

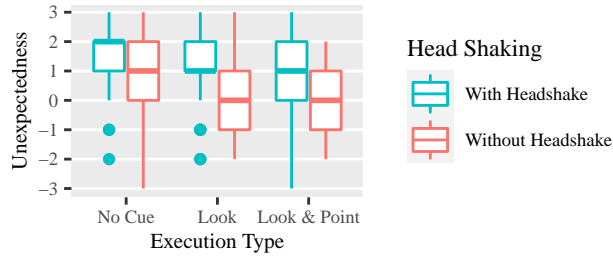


Figure 4: Boxplot of Unexpectedness responses. All robot behaviors were rated unexpected except for Look without Headshake (the middle red box) and Look & Point without Headshake (the right red box).

weak evidence¹³ in the Look with Headshake condition (median scores in the top row in Figure 3).

We also performed post-hoc pairwise comparisons using Tukey’s test with Holm-Bonferroni correction ($H_0 : \mu_i = \mu_j$). Without Headshake, statistically significant differences were found between Look & Point and No Cue ($0.80 \pm 0.24, p < 0.01$), and Look and No Cue ($0.80 \pm 0.24, p < 0.01$). With Headshake, we only found the statistically significant difference between Look & Point and No Cue ($0.85 \pm 0.24, p < 0.01$) conditions.

The pairwise comparisons suggest the No Cue execution was more unexpected than the Look & Point and Look conditions. However, no statistically significant differences between Look & Point and Look execution conditions were found in either Headshake condition.

In summary, **H3.1** which states that all robot behavior would be considered unexpected was partially supported. For the without Headshake conditions, participants reported that the robot’s behavior in only the No Cue condition was unexpected but rated it more neutral in the Look and Look & Point conditions. The robot’s behavior was rated as unexpected across Headshake conditions, however.

3.4.2 H3.1, H3.2 & H3.3: Causal information, Headshake, and need for explanations

Similar to the unexpectedness responses, we calculated an average score for responses to the need for explanation items. Cronbach’s alpha for this subscale was 0.74. Responses are plotted in

¹³This effect becomes non-significant in our replicated study.



Figure 5: The distribution of Need responses with median lines and estimated marginal means, showing that robot behavior should be explained in all conditions. No statistical significance was found in post-hoc pairwise comparisons.

Figure 5 and 6. We also used the between-subjects factorial ANOVA to analyze the data. Again, we did not find a statistically significant interaction between *Headshake* and *Execution Type*, but found statistically significant main effects for *Headshake* ($F(1, 360) = 14.99, p < 0.0001$) and *Execution Type* ($F(2, 360) = 3.31, p < 0.05$) for explanation scores.

Again, before we conducted pairwise comparisons across conditions, we used the ANOVA model to calculate estimated marginal means of all conditions and performed multiple comparisons with Holm-Bonferroni correction [89] to test whether these means significantly deviate from 0 ($H_0 : \mu = 0$). Results show statistical significance across all conditions ($p < 0.001$):

- Without headshake:
 - Look & Point: $0.75 \pm 0.18, p < 0.0001$
 - Look: $0.62 \pm 0.18, p < 0.001$
 - No Cue: $1.18 \pm 0.18, p < 0.0001$

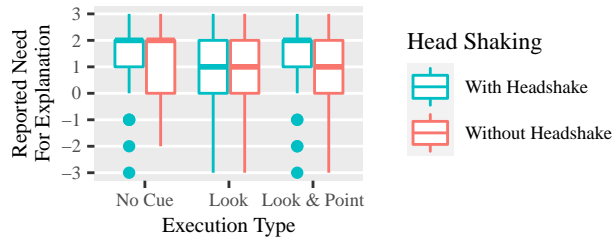


Figure 6: Boxplot of the Need for explanation responses. Participants in all conditions agreed that robot behavior should be explained. No significant differences were found pairwise between conditions.

- With headshake:
 - Look & Point: $1.43 \pm 0.18, p < 0.0001$
 - Look: $1.25 \pm 0.18, p < 0.0001$
 - No Cue: $1.61 \pm 0.18, p < 0.0001$

This result suggests that people want the robot to explain and that the robot should explain its behavior across all conditions, even when additional arm movement (Look & Point), head turning (Look), and Headshake cues are included.

Pairwise comparisons were also conducted with Tukey’s test ($H_0 : \mu_i = \mu_j$), not revealing any statistically significant differences between the execution type conditions. This suggests that people want the robot to explain in all conditions equally.

The analysis supports part of **H3.1** that robot behavior should be explained, even when robot behavior includes non-verbal motion cues. Together with responses to the Unexpected items, **H3.1** was **partially supported**: when the robot’s behavior is deemed neutrally unexpected, the robot should explain its behavior. However, **H3.2** was not supported. As more causal information was added across the conditions from No Cue to Look & Point, the perceived need for explanation did not drop accordingly. **H3.3** was also not supported: The pairwise comparisons showed no statistically significant differences between with Headshake and without Headshake groups, suggesting that the addition of the Headshake cue did not result in a less perceived need for an explanation

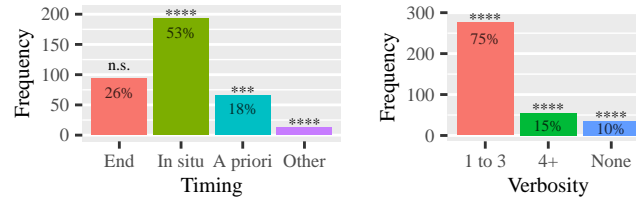


Figure 7: Timing and verbosity preferences. Around half participants prefer the robot to explain *in situ* and most (75%) participants are willing to ask only a few clarifying follow-up questions.

from the robot.

3.4.3 H3.4: Explanation timing

We performed a chi-square goodness-of-fit test on the responses to the multiple-choice timing question, which revealed statistical significance ($p < 0.0001$). Post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons were performed and show significant differences between *in situ* ($p < 0.0001$), *a priori* ($p < 0.001$) and the other response options ($p < 0.0001$), except for explaining at the end ($p = 0.76$). Among the 13 participants who elaborated their choice for “other”, four participants expressed that they wanted a *running commentary* from the robot from the beginning of its action, which is different from asking *in situ* questions. All other comments were either isolated or very similar to the other three choices.

Thus, **H3.4** was not fully supported. As shown in Figure 7 left, approximately half (193/366, 53%) of the participants wanted the robot to explain *in situ* as unexpected things happened, and only 66 (18%) participants wanted explanations from the robot before something unexpected happens. However, the binomial test shows that the choice of subsequent explanations may have happened by chance: whether more or fewer people wanted explanations after the robot’s behavior remains unknown.

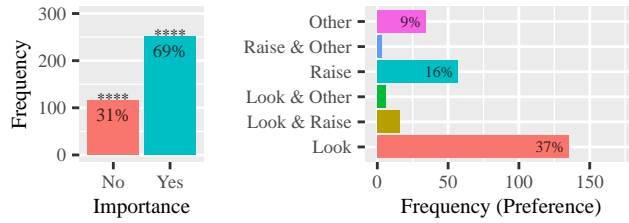


Figure 8: Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). Most participants (69%) agree it is important for the robot to engage with them before explaining. Slightly over one-third of all participants prefer looking at them to get their attention.

3.4.4 H3.5: Engagement importance/preference

A chi-square test was run on the engagement importance true/false responses and indicated that the proportion of “false” responses (115, 31%) to the item asking about whether it was important for the robot to get the participant’s attention before giving an explanation was significantly lower than “true” responses (251, 69%; $p < 0.0001$), as shown in Figure 8 left. Thus, **H3.5** was supported: it is important for the robot to get one’s attention before explaining.

Regarding the preference for *how* the robot should get the attention of humans (Figure 8 right), a multinomial goodness-of-fit test was performed and shows statistical significance ($p < 0.0001$). Post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons were performed and revealed significant differences between all choices ($p < 0.0001$ for all) but not for Raising the volume ($p = 0.62$), which suggests people may have selected this choice at random. In summary, more than one-third of participants (135, 37%) preferred the robot to look at them to get their attention, and only 57 participants (16%) preferred the robot to raise its volume. Interestingly, no participants chose the combined option to “Look, Raise & Other”.

Among the 43 participants who elaborated their answers about the “Other” choice, 30 participants wanted the robot to address them by name or title or with words such as “hey” or “excuse me”; 7 participants wanted a beep sound while the other responses were isolated.

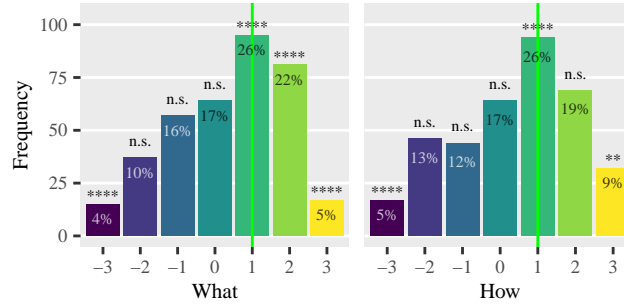


Figure 9: Robot vs. human explanations in what and how (green line indicates median). Half participants agreed on no differences in both. Please see Section 3.4.5 for more details.

3.4.5 H3.6: Similarity to human explanation

For the two questions in this scale, Cronbach’s alpha reports 0.49, suggesting those two questions represent two independent subscales. Upon further reflection, we realized that the first question was asking about the difference in *what* to explain (i.e., “there should be no difference between what a robot says to explain its behavior and what a person would say to explain the same behavior.”) The second question was asking about the difference in *how* to explain (i.e., “if a person did what the robot did, they should both explain the same behavior in the same way.”)

Thus, to test the responses, we ran a chi-square goodness-of-fit test on the responses to both items independently. The chi-square test for the first question indicated a statistical significance ($\chi^2(6) = 108.58, p < 0.0001$). Post-hoc binomial tests with Holm-Bonferroni correction show statistically significant differences between Likert responses -3 ($p < 0.0001$), 1 ($p < 0.0001$), 2 ($p < 0.0001$), 3 ($p < 0.0001$) but not in -2 ($p = 0.06$), -1 ($p = 0.46$) and 0 ($p = 0.17$).

We also ran the tests on the responses to the second question. The chi-square test indicated a statistical significance ($\chi^2(6) = 75, p < 0.0001$). Post-hoc comparisons show statistically significant differences in Likert responses -3 ($p < 0.0001$), 1 ($p < 0.0001$), and 3 ($p < 0.01$) but not in -2 ($p = 0.46$), -1 ($p = 0.46$), 0 ($p = 0.26$) and 2 ($p < 0.06$).

Thus, **H3.6** was partially supported. Approximately half of the participants agreed that the robot should explain the behavior by saying the same thing as a human would (53%, Figure 9 left).

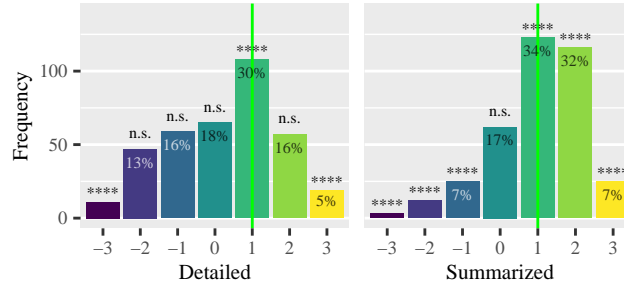


Figure 10: Two summarization aspects (green lines indicate median values). *Detailed*: While 35% participants (1 & 3) agreed explanations should be detailed, almost half of the responses (-2, -1, 0, 2) may happen at random. *Summarized*: 72% participants preferred explanations to be concise.

But only one-third of participants agreed they should explain in the same way (35%, Figure 9 right). More than one-third of participants who indicated that the robot should explain in a different way than a human may have happened by chance (43% and 42% respectively).

3.4.6 H3.7, H3.8: Detail, summarization and follow up questions

Cronbach’s alpha reports -0.57 on the three summarization questions with the first being reversed, but 0.65 (acceptable with a large sample) with the first question dropped, suggesting the first question may be measuring an independent metric – detailed – and the last two questions for another metric – summarized.

Explanation detail Similar to the human-robot difference question, we ran a chi-square goodness-of-fit test on the responses to the first question, indicating a statistically significant difference ($\chi^2(6) = 118.07, p < 0.0001$) between responses. Post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons show statistically significant differences in -3 ($p < 0.0001$), 1 ($p < 0.0001$), 3 ($p < 0.0001$) but not in -2 ($p = 0.99$), -1 ($p = 0.99$), 0 ($p = 0.28$) and 2 ($p = 0.99$). Thus, at least 35% agreed that explanations should be very detailed, shown in Figure 10 left. However, the fact that there were four Likert scale points not being statistically significant may imply that almost half (47%) of participants may not have been certain about the need for detailed expla-

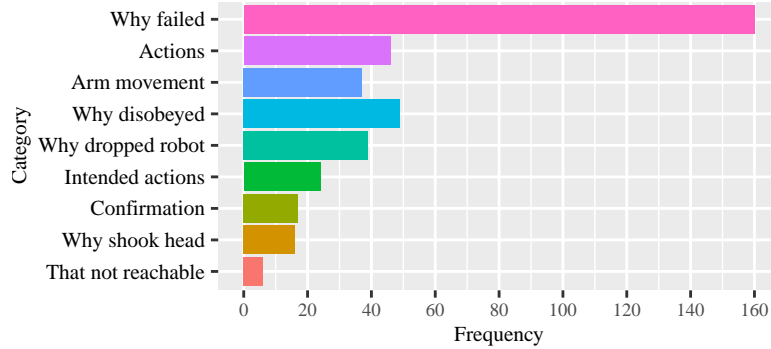


Figure 11: The most wanted robot explanations (Top categories regarding explanation content). The top one is why the robot failed to hand the cup.

nations.

Explanation summary We calculated an unweighted average score from the responses to the last two questions to achieve a composite score. A chi-square goodness-of-fit test was performed and indicates a statistically significant difference ($\chi^2(6) = 281.06, p < 0.0001$) and Post-hoc binomial tests with Holm-Bonferroni correction show statistically significant differences in all ($p < 0.0001$) but not in 0 ($p = 0.16$). In summary, 72% participants preferred explanations to be concise, with only 11% disagreeing, shown in the right of Figure 10.

Level of verbosity We performed a multinomial goodness-of-fit test on the responses, which reveals statistical significance ($p < 0.0001$). Post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons were performed and show significant differences in all levels ($p < 0.0001$). As shown in Figure 7 right, 75% preferred “Only A Few (1 to 3)”.

Thus **H3.7** and **H3.8** were supported. Participants preferred that the robot’s provided explanation be a summary as opposed to being detailed, and they preferred fewer follow-up questions as opposed to more.

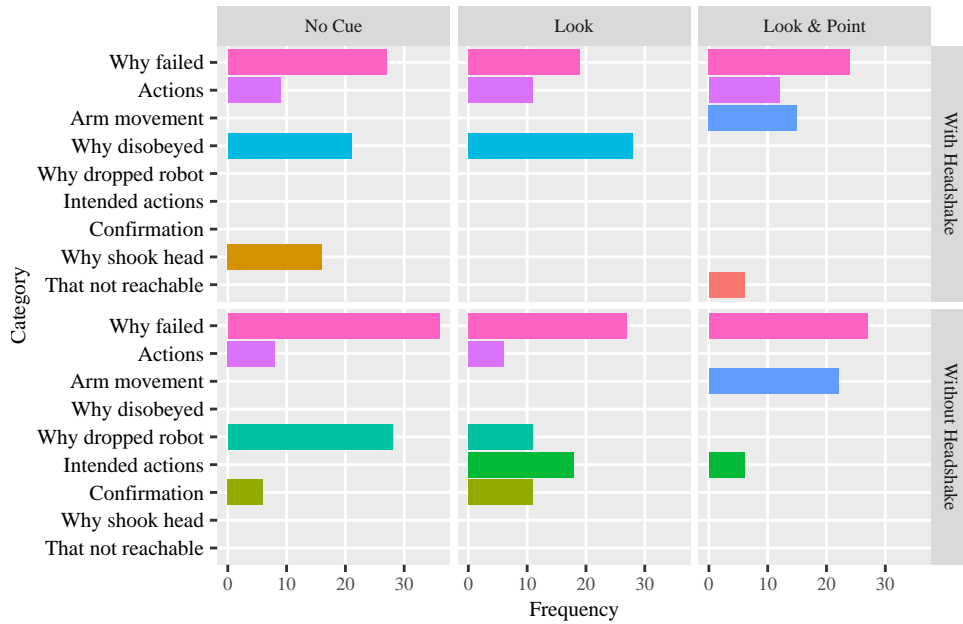


Figure 12: The most wanted robot explanations across conditions (Most frequent categories regarding explanation content across conditions). An explanation for why the robot failed to pass the cup is common in all conditions. Please see Section 3.4.7 for other categories.

3.4.7 Additional analyses: Explanation content

We coded the open-ended comments on what should be explained and grouped them into multiple categories. An independent coder coded a random sample of 10% of the participant data and the experimenter coded all responses. After merging codes with similar meanings, we achieved a Cohen’s κ value of 0.78, considered substantial agreement between raters by [101]. Figure 12 shows those conditions in which at least 6 participants (approximately 10% of participants in a given condition) endorsed the coded category. Figure 11 shows the data without conditions.

Common to all conditions except for the Look with Headshake condition (31% participants), around half of the participants (39% to 59%) wanted the robot to explain **why it failed** to pass the cup (pink bars in Figure 12): No Cue without Headshake: 59%, No Cue with Headshake: 44%, Look without Headshake: 44%, Look & Point without Headshake: 44%, Look & Point with Headshake 39%).

For the Look with Headshake condition (top middle in Figure 12), people were more spe-

cific, with 46% participants asking **why the robot disobeyed** them by shaking its head (purple in Figure 12). This perception of defiance also held for 34% of the participants who were in the No Cue with Headshake condition (top left in Figure 12).

Perceptions of the robot acting defiantly were not shown for the Look & Point condition because arm movement likely indicated that the robot obeyed, but 36% in the Look & Point without Headshake (22) and 25% in the Look & Point with Headshake (15) conditions were confused about **arm movement intention** (cyan in Figure 12) and only 6 participants in total (Look & Point with Headshake) explicitly expressed a desire for the robot to say **that it could not reach the cup**.

For No Cue execution conditions, **why it dropped** the small toy robot that Baxter was holding onto the desk (sky-blue in Figure 12) and **why it shook its head** (brown) were mostly questioned in the No Cue without Headshake condition (46%, 28) and the No Cue with Headshake condition (26%, 16) respectively. With an additional cue like head-turning or arm movement but not Headshake, the percentage of participants who wanted to know why the small toy robot is dropped decreased to 18% (11) in the Look without Headshake condition and disappeared in the Look & Point without Headshake condition. With Headshake (No Cue with Headshake, Look with Headshake, Look & Point with Headshake), no participants wanted to know why the robot dropped the small toy robot Baxter was holding onto the desk. Except for the No Cue with Headshake condition where Headshake was the only motion cue, no participants wanted to know why the robot shook its head.

For general explanations, around 10% participants wanted explanations for the robot's **actions**, mostly current actions, in each condition (No Cue without Headshake: 13%, 8; No Cue with Headshake: 15%, 9; Look without Headshake: 10%, 6; Look with Headshake: 18%, 11; Look & Point with Headshake: 20%, 12) except for the Look & Point without Headshake condition where explanations for the arm movement intention were more desired. Being related, participants in the Look without Headshake (30%, 18) and Look & Point without Headshake (10%, 6) conditions wanted explanations for the robot's **intended actions**. Lastly, when there were no motion cues (No

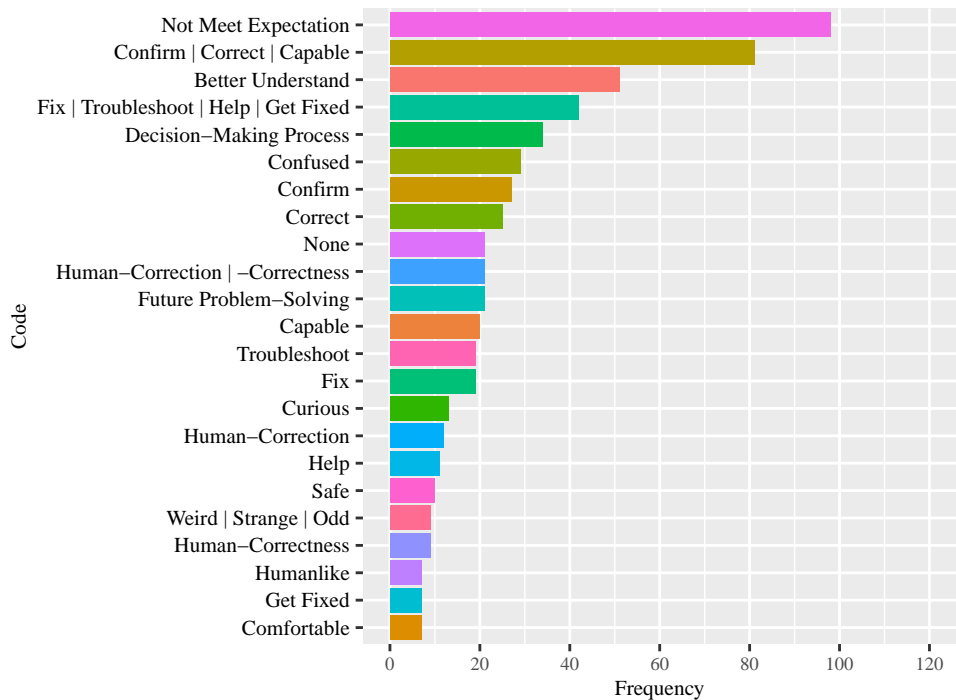


Figure 13: Most frequent coded responses for why participants want the robot to explain (i.e., explanation reasoning). Two of them are endorsed by more than 20% of participants: the robot should explain because its behavior does not meet their expectations, and in order to confirm the robot will do the task, will do it correctly, and whether it is capable of finishing the task.

Cue without Headshake) or just head turning (Look without Headshake), 10% (6) and 18% (11) wanted **confirmation** of the participant’s request from the robot respectively.

3.4.8 Additional analyses: Reasoning behind explanation content

Similarly, we also coded participant comments on why the robot should explain. This question was asked immediately after the explanation content question.

Figure 13 shows the top 23 most frequently coded responses to the “Why the robot should explain” open-ended question, that appeared more than 6 times, i.e., 10% of participants. Figure 14 shows the same data but across conditions.

Ninety-eight (26.8%) participants wanted the robot to explain because the handover failure did **not meet their expectations**. While there were around 20 cases for No Cue conditions (both

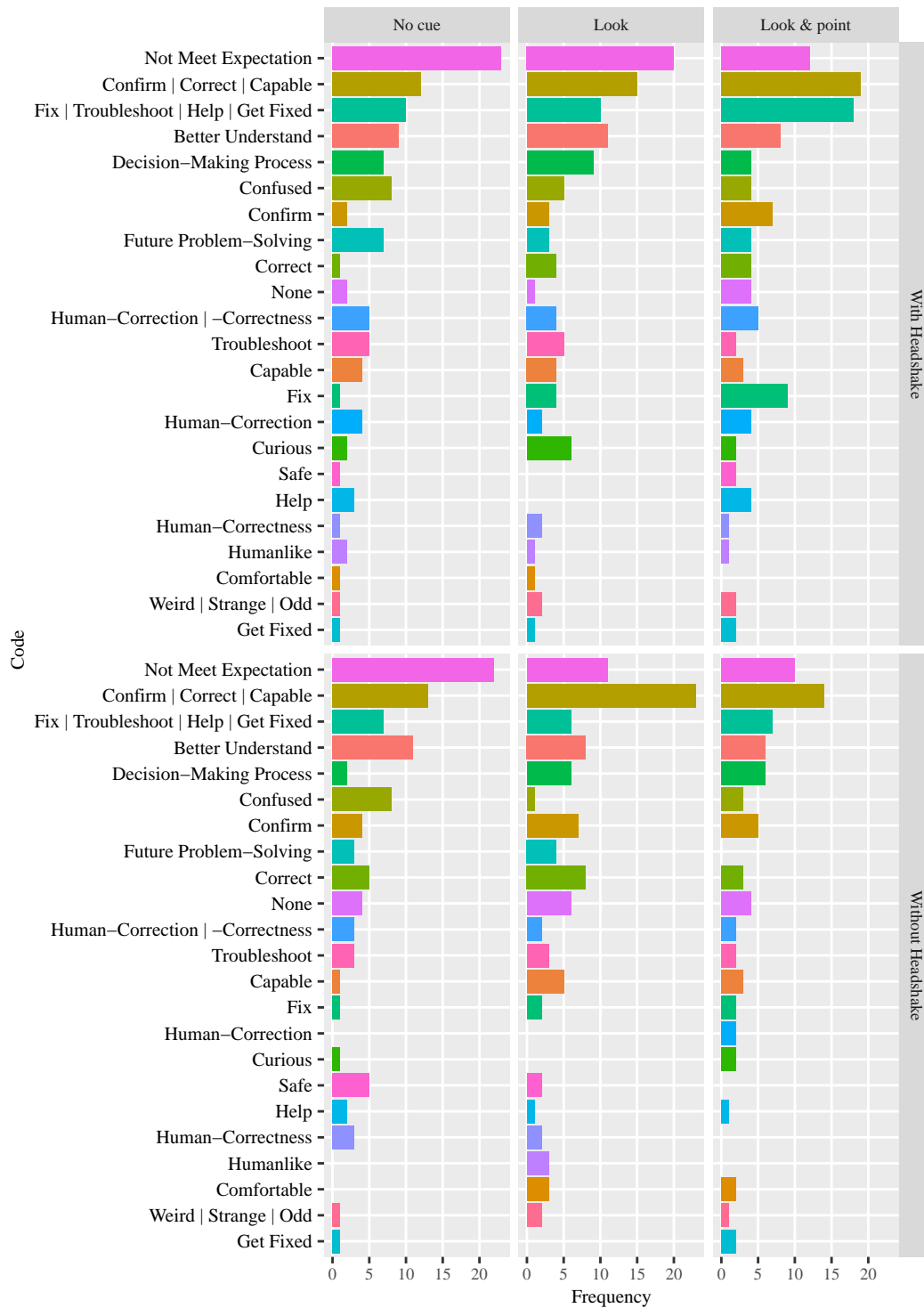


Figure 14: Top codes for explanation reasoning across conditions. Please see Section 3.4.8 for a detailed analysis.

without Headshake and with Headshake) and the Look condition with Headshake, there are only around 10 for the Look condition without Headshake and both Look & Point conditions.

By comparing the No Cue and Look conditions, we found that participants expected the robot to turn its head towards the cup (Look without Headshake) but without a Headshake, which explains why the count for the Look without Headshake condition is lower. The takeaway here is that **participants set expectations of successful handovers from the robot after their request, and the robot should have explained when it could not meet those expectations.**

By comparing the Look and Look & Point conditions, the additional arm movement does not lead to any change when there is no headshake, but the count reduces by half with a headshake. However, the reason why more participants want the robot to explain itself is to fix the robot or the second composite code, revealing how problems are perceived. The takeaway here is that **when the robot cannot complete the task yet exhibits some unclear behaviors without explanation, participants interpreted them as problems and that the robot needed to be fixed**¹⁴.

For other provided reasons, the differences are not large across conditions, usually within 5–10, so we will not discuss them per condition below.

As seen in Figure 13, the second most frequent response by 81 participants (22.1%) was a composite one: *Confirm | Correct | Capable*, indicating that the robot should explain in order to confirm it will do the task, will do it correctly, and whether it is capable of finishing the task. Around 51 participants (13.9%) expressed general reasoning: robot explanation helps them better understand the robot. Interestingly, in the fourth composite response: *Fix | Troubleshoot | Help | Get Fixed*, 42 participants (11.5%) expressed interest in solving the problem of the robot, either by themselves or by contacting the manufacturer of the robot. Related to this, we found 21 participants (5.7%) who stated that if the robot explained that the participant was the cause of the problem that they would like to adjust their own actions to help the robot complete its tasks, which are coded as *Human-Correction | -Correctness*.

¹⁴Due to the open-ended nature, we did not find the same evidence in the 2020 Study.

Due to the open-ended nature of the question, all other themes found in the responses were fragmented and limited to less than 10% of participants. Some interesting reasons included understanding the decision-making process of the robot (34 participants, 9.3%), and being able to solve future problems after understanding current problems (21 participants, 5.7%). If these options are given explicitly, e.g., in a forced-choice question, more participants may choose them.

3.5 Replicating The Study To Verify Results

To validate the robustness of the results, we conducted a strict replication of the study. We launched the study again on MTurk, following the identical procedures reported above. The original study was conducted in August 2019 (“2019 Study”), and the replication was conducted 15 months later in November 2020 (“2020 Study”). Due to the ongoing COVID-19 pandemic, our ability to conduct in-person human subjects studies has been limited as in many other research labs. We thus were unable to replicate the study in a laboratory setting.

We aimed to recruit an identically sized sample as the 2019 Study from MTurk following the same procedure. For the 2020 Study, MTurk workers who participated in the 2019 Study were excluded from participation. While reviewing the responses to the attention check questions, we also found that 38 participants gave suspect answers to the question asking about explanation content. Specifically, they provided a full or partial copy of the definition of what a robot is and how it works from top web search returns including Wikipedia. We decided to exclude these participants’ responses, as the answer was non-responsive and suggested either the person was not paying close attention or a bot may have been filling out the survey. Given this change to our attention check questions, we went back to the data from the 2019 Study, as we described before its analysis.

After employing the same screening procedures as the 2019 Study, including removing participants who gave suspect answers noted above, we trimmed the dataset to provide an equal number of participants in each of the 6 experimental conditions and to match the size of the 2019

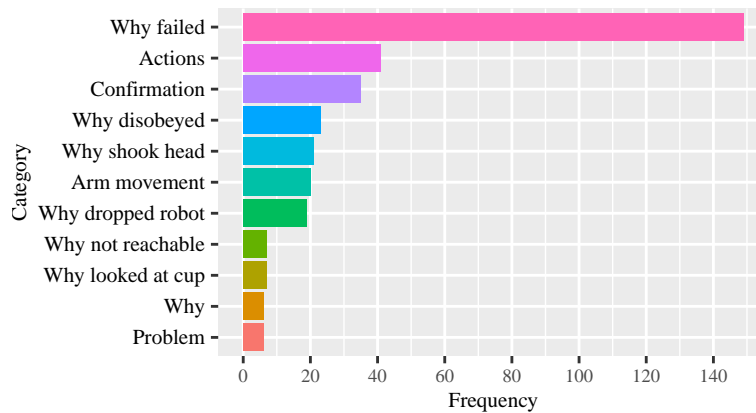


Figure 15: Top categories regarding explanation content (replication results from the 2020 Study). The top one remains unchanged.

Study sample. Due to this process, we again had data from $N = 366$ participants, which is included in subsequent analyses.

The demographics of the final sample for the 2020 Study were very similar to the 2019 Study, with 206 males, 157 females, 1 participant who preferred not to say, and 2 transgender people; with ages ranging from 20–69, $M = 36$, $median = 33$, $skewness = 1.05$. 101 participants (28%) agreed with the statement, “I have experience with robots,” 191 disagreed (52%), and 74 (20%) responded that they neither agreed nor disagreed.

After following identical analysis procedures as with the 2019 Study data, we found that the 2020 Study replicated all of the 2019 Study findings except for the Unexpectedness score of the Look & Point with Headshake condition. As seen in the top right subfigure in both Figure 64 and Figure 65, we no longer found any significant difference (was $p < 0.01$ for the 2019 Study) between the mean estimate of responses and 0 (used to code neutral responses).

We placed all of the figures for the 2020 Study and the 2019 Study side by side in Appendix A for easy comparison and to document the few minor statistical significance changes in findings regarding Hypotheses H3.1-H3.8 between the two studies.

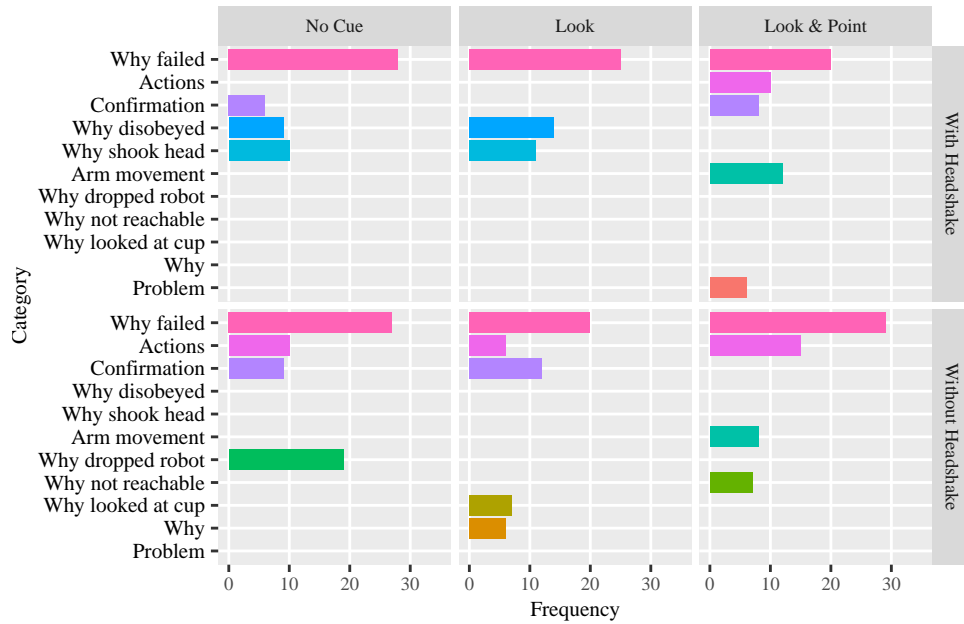


Figure 16: Most frequent categories regarding explanation content across conditions (replication results from the 2020 Study). Still, the explanation for why the robot failed to pass the cup is common in all conditions. Please see Section 3.5.1 for other categories.

3.5.1 Additional analyses: Explanation content

As shown in Figure 15 and 16, we were able to draw roughly the same conclusions for explanation content as in the 2019 Study (Section 3.4.7).

The top explanation that the participants explicitly wanted the robot to provide remained **why it failed** to pass the cup (40%). The same as in the 2019 Study, and this finding was common to all conditions, with a maximum difference of 9 participants (No Cue with Headshake vs. Look & Point with Headshake): No Cue without Headshake: 45%, No Cue with Headshake: 47%, Look without Headshake: 33%, Look with Headshake: 42%, Look & Point without Headshake: 48%, Look & Point with Headshake 33%.

Regarding the explanation for **why the robot disobeyed** by shaking its head, similar to the 2019 Study, it only appeared for the No Cue with Headshake and Look with Headshake conditions.

With the data about compliance above, the conclusion of our previous analysis still holds: “Perceptions of the robot acting defiantly were not shown for the Look & Point condition because

arm movement likely indicated that the robot obeyed.” There were still participants who explicitly expressed that they would like explanations about **the arm movement intention**, although the number dropped from 22 and 15 participants to 12 and 8 respectively.

For No Cue without Headshake condition, **why it dropped** the small toy robot that Baxter was holding onto the desk remained to be mostly questioned (19, 32%). While **Why shook head** is still the most questioned for No Cue with Headshake condition, its number has dropped from 16 to 10 participants. One new finding is that in the Look with Headshake condition, 11 participants (18%) were confused about why the robot shook its head. The conclusions given in the second-to-last paragraph of Section 3.4.7 still largely hold.

For general explanations, there were around 10% of participants who wanted explanations for the robot’s **actions**. These participants were roughly evenly distributed in most conditions, but not in the No Cue with Headshake and Look with Headshake conditions (Figure 16). This differs from the 2019 Study in which participants did not want an explanation for the robot’s actions in only the Look & Point without Headshake condition (Figure 12).

For **confirmation**, there were still approximately the same number of participants who wanted a confirmation of their request from the robot: no motion cues (No Cue without Headshake, 12 participants in the 2020 Study vs. 6 in the 2019 Study) or just head turning (Look without Headshake, 14 in the 2020 Study vs. 11 in the 2019 Study).

3.5.2 Additional analyses: Reasoning behind explanation content

As shown in Figure 17, that the robot did **not meet their expectations** remains to be the top reason behind what to explain, which increased from 98 participants (26.8%) to 113 (30.9%). Across conditions (Figure 18), there were still only around 15 cases in the Look without Headshake and more than 20 cases in the No Cue and Look with Headshake conditions. The conclusions given in the fourth paragraph in Section 3.4.8 still hold.

The composite code “Confirm — Correct — Capable” was still the second-most frequent

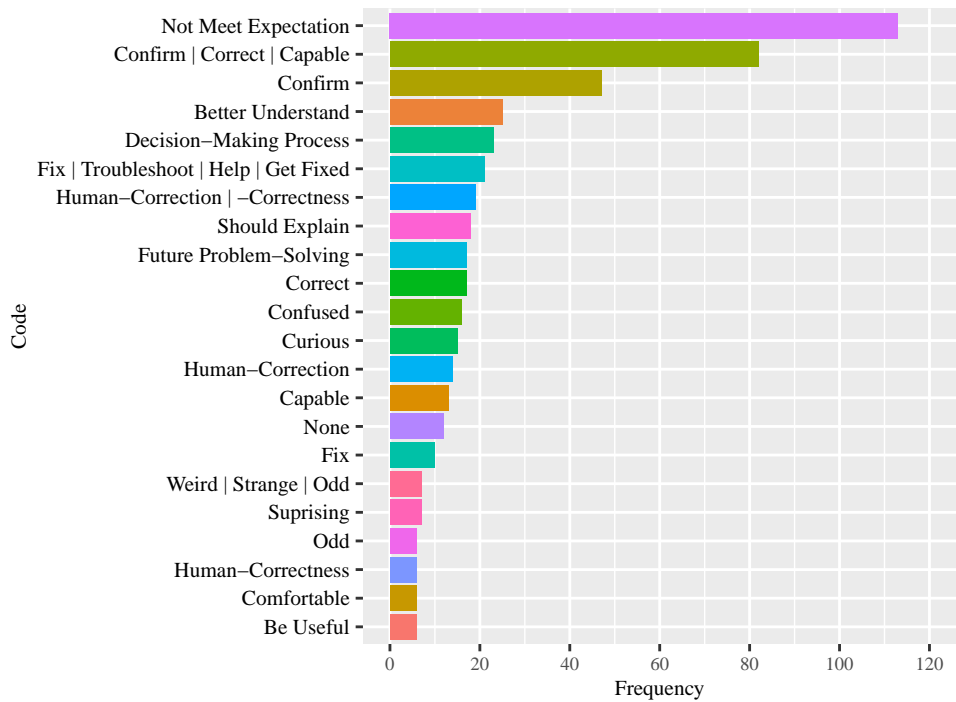


Figure 17: Most frequent coded responses for explanation reasoning (replication results from the 2020 Study). The top two remain unchanged and are still endorsed by more than 20% participants. For the changes from the 2019 Study, please see Section 3.5.2.

reason, with roughly the same number of participants in the 2019 Study. Forty-seven vs. 27 participants expressed that they wanted to get confirmation from the robot out of an explanation. The codes “Better Understand” and “Fix — Troubleshoot — Help — Get Fixed” dropped 5.4% from 40 participants to 20. Due to the open-ended nature, we did not find the same conclusion earlier about getting the robot fixed. The frequency values for other codes from the 2019 Study are within around 10 participants each.

3.6 Discussion

Although the data only partially supported H3.1: the Look without Headshake and the Look & Point without Headshake conditions were considered to be unexpected by participants; we found that outside of H3.1 on the unexpectedness metric, adding additional non-verbal cues to increase causal information was not always helpful for meeting participant expectations. Comparing verti-

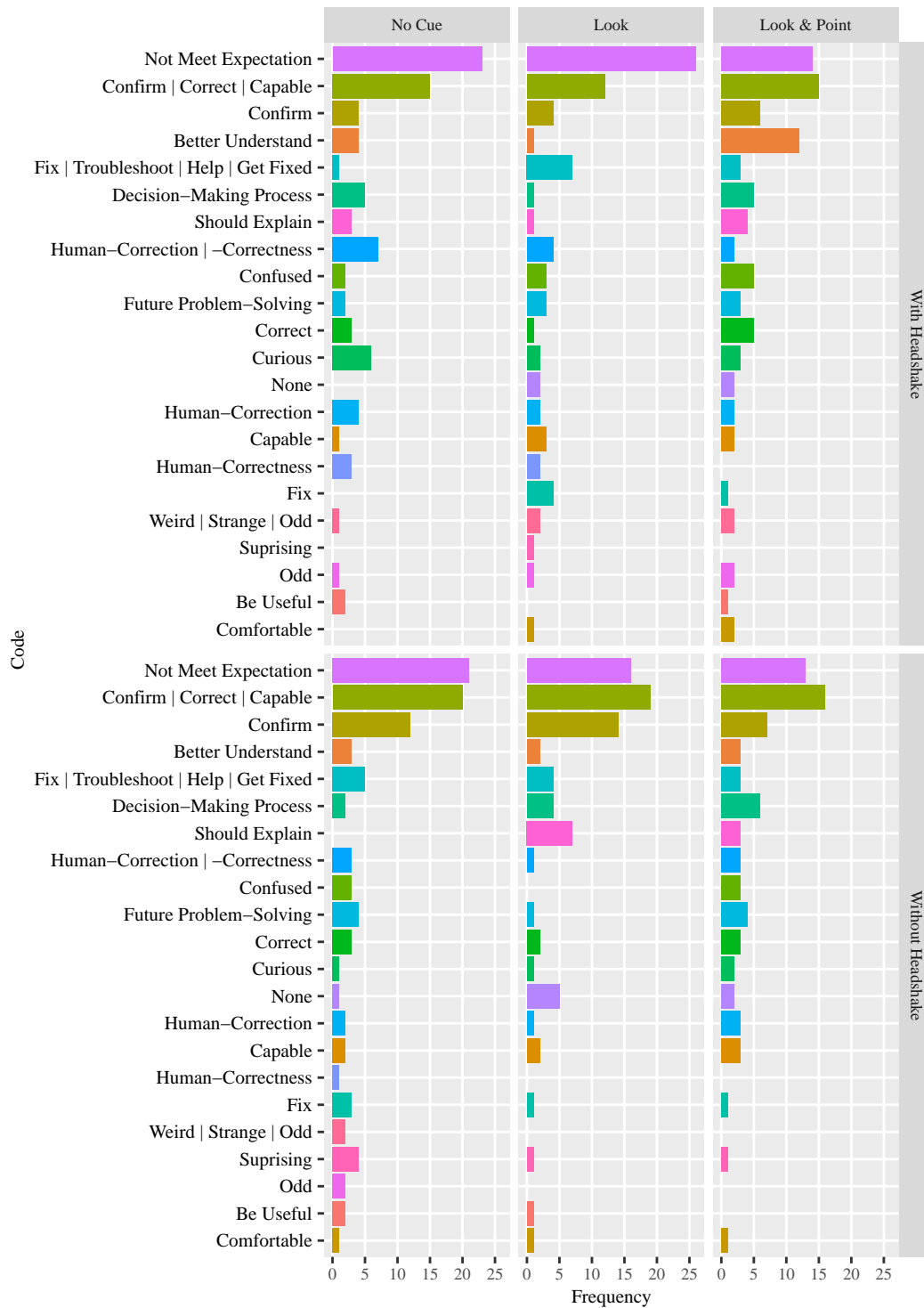


Figure 18: Top codes for explanation reasoning across conditions (replication results from the 2020 Study). please see Section 3.5.2 for a detailed analysis.

Table 3: Summary of evidence or lack thereof for hypotheses (includes data from both the 2019 Study and the 2020 Study)

| Hypothesis | Result |
|--|---|
| H3.1. Robot behaviors need to be explained. In general, robot behavior will be considered unexpected to people, and there will be a desire for it to be explained. | <i>Partially supported.</i> Results show that the robot should explain even when participants rank its behavior as neither expected nor unexpected (i.e., neutral) in the Look without Headshake and Look & Point without Headshake conditions. |
| H3.2. As more causal information about robot behavior is provided, there will be less need for an explanation from the robot. | <i>Not supported.</i> The perceived need for explanation did not drop with more causal information. |
| H3.3. Adding a headshake to the robot's explanation will result in less need for an explanation. | <i>Not supported.</i> The addition of the Headshake cue did not result in less perceived need for explanation. |
| H3.4. Explanations offered at multiple points in time will be desirable. | <i>Not fully supported.</i> More participants (53%) preferred in-situ explanations, fewer (18%) preferred a-priori, yet no statistical difference was found for explanations in the end. |
| H3.5. Engagement prior to providing an explanation will be important. | <i>Supported.</i> |
| H3.6. Similarity to human explanations will be expected. | <i>Partially supported.</i> Half of the participants agreed that robots and humans should say the same thing, but only one-third of participants agreed they should explain in the same way. |
| H3.7. Summarization will be preferred. | <i>Supported.</i> |
| H3.8. Fewer number of follow-ups will be preferred. | <i>Supported.</i> |

cally in Figure 64 for the 2019 Study and Figure 65 for the 2020 Study, the addition of the Headshake without any verbal explanation made the robot's behavior unexpected (Look with Headshake vs. Look without Headshake in the 2019 Study and the 2020 Study), equally unexpected (No Cue without Headshake vs. No Cue with Headshake) or more unexpected only in the 2019 Study (Look & Point without Headshake vs. Look & Point with Headshake).

Although the addition of head-turning in the Look condition did decrease the unexpectedness ratings over the No Cue condition in the 2019 Study but not in the 2020 Study (compare the left two columns of Figure 64), the inclusion of arm motion in the Look & Point condition failed to make the robot's behavior less unexpected than the Look condition – the unexpectedness either increased in the 2019 Study (compare the right two columns of Figure 64) or was not significantly significant in the 2020 Study (compare the right two columns of Figure 65).

In contrast to H3.2, we found that in both the 2019 Study and the 2020 Study, increasing causal information by adding more non-verbal motion cues did not increase the need for robot explanations. Rather, the need was always desired regardless of what non-verbal cues were being used (partially confirms H3.1). Together with the explanation content responses, more non-verbal cues such as head shaking or arm movement did not seem to decrease the need, but simply led participants to seek additional specific explanations for those cues.

The assumption that the addition of a headshake would lead to less need for explanation, as hypothesized in H3.3, did not hold. In the comments on explanation content, participants reported being confused about the robot's headshaking behavior: instead of perceiving the robot as unable to take the cup, the headshake was considered as a sign that the robot was disobeying many of the participants (the most frequent category in the 2019 Study and second-most in the 2020 Study), when additional arm movement was not present. This confusion may explain why H3.3 was not supported. Even though we limited our recruitment on MTurk to people living in the United States, cultural difference, especially concerning the meaning of a headshake, may be another factor affecting the perception.

Confusion also existed for the Look & Point execution conditions, with some participants not understanding the intention of the robot's arm movement. This suggests that non-verbal behaviors such as head-shaking and artificial yet unpredictable human non-verbal behaviors such as repeated reaching arm movement without explanation may fall short of being clear methods for trying to provide in situ motion-based implicit explanations.

For the properties of robot explicit explanations, participants wanted robots to explain as unexpected things happened. However, we saw explicit requests for a running commentary of the robot's behavior by 4 participants in the 2019 Study and 3 participants in the 2020 Study. Similarly, while participants preferred the robot to get their attention by looking at them, 30 participants in the 2019 Study and 19 in the 2020 Study explicitly opted for the robot to address them specifically.

Surprisingly, results showed that there was no desired difference between robot and human explanations, rejecting H3.6. However, most non-agreement responses to the difference between robot and human explanation items were not of statistical significance (Figure 88 for the 2019 Study and Figure 89 for the 2020 Study), nor were most participants' responses to the detailed aspect in summarization (Figure 91 for the 2019 Study and Figure 92 for the 2020 Study). Also, the distributions shown in Figure 88 for the 2019 Study and Figure 88 for the 2020 Study are very similar, suggesting that people may not differentiate what and how robots explain. Additional investigation, with multiple questions to improve reliability, would be needed to lend more clarity to these findings. In addition to the arm and head movement, (i.e., non-verbal motion cues), future work should also include exploration to learn if eye gaze or facial expression would be unexpected and how they would affect perceptions of robot explanation.

3.7 Limitations and Future Work

In the experiment, we focused on robot explanations for failures that occur shortly after a person's request, using the example of handover interaction. This timing is important because early failures prevent the whole process from happening and early failures have been shown to decrease a per-

son's trust in a robot system [46]. While seemingly simple, the handover interaction was chosen as it is expected to be a frequent interaction among humans working in physical proximity to robots. However, due to the timing of our work and the global pandemic, no in-person or physical interaction was involved in this study (i.e., transferring the object to participants). Further investigation is needed for physical proximity and contact between humans and robots, and for different contexts and tasks other than handovers, likely using in-person studies, although this work has demonstrated that consistent results can be attained through online studies.

In addition, while we aimed to have our findings generalize to the general public, preferences are inherently subjective, influenced by culture, and individual differences exist. Thus, adaptive explanation to individuals is of interest and needs more investigation but is out of the scope of this work. As we have only used the Baxter robot with a slightly smiling face and moderate human-likeness, future work could further investigate how different robot designs impact participant expectations for robot explanation.

Knowing people's preferences for robot explanations paves the way to explanation generation. In our parallel work [82], we proposed such algorithms to generate shallow explanations for only a few follow-up questions, informed by the summarization and verbosity preferences. Currently, we are investigating the effects of verbal explanation mixed or not mixed with non-verbal projection mapping [83] cues, informed by the engagement and timing preferences, i.e., addressing humans in situ.

3.8 Conclusions

We have investigated desired robot explanations when coupled with non-verbal motion cues. Results suggest that robot explanations are needed even when non-verbal cues are present, in most cases where the robot is unable to perform a task. The robot needs to get the attention of the person and then concisely explain why it failed and its other behaviors in situ, in a similar way to what humans would expect other humans to do. We are able to make the same conclusion in a strict

replication study after 15 months, providing stronger evidence for our findings.

With these results, the next chapter will detail how we generate concise explanations and follow-up explanations to answer a few questions.

4 Explanation Generation Using Behavior Trees¹⁵

4.1 Introduction

As robots are pushed by researchers and the industry to complete more complex tasks, improving the understanding of a robot’s behaviors is becoming increasingly important. Prior work in human-robot interaction (HRI) has shown that improving understanding of a robot makes it more trustworthy [46] and more efficient [7]. To increase human understanding of robots, HRI researchers have mostly explored non-verbal physical behavior such as arm movement [56, 100] and eye gaze [118]. Such non-verbal behaviors can help people to anticipate a robot’s actions [102], but do not provide insight into *why* the robot chose those actions. Direct explanations of why certain behaviors occurred can further improve one’s prediction of behaviors [108]. Thus, robots need to explicitly explain their own behavior, similar to verbal human explanations.

Previously, we proposed a holistic blueprint of a robot explanation generation system consisting of three components: state summarization, data storage and querying, and a human interface [80]. The state summarization component is responsible for generating varying levels of summaries, either manually or automatically from different robot states, while performing tasks or from the stored states in introspection. In this work, we focus on the manual generation of high-level explanations, for situations where robot programmers want to consider possible explanations to create more transparent robot systems as they program the functionality of the system.

We explore action sequence methods for task specification and execution as the foundation for robot explanation generation. Specifically, we base our work on *Behavior Trees* (BTs), which is a tree structure that encapsulate behavior by control nodes that contain child execution nodes. BTs [40] are a popular way to model the behavior of AI agents in the gaming industry (e.g., [103, 141]), but have also gained momentum in robotics, where they are used for end-user robot programming [133], deployed on a Rethink Robotics Sawyer robot [40], and applied to Learning

¹⁵This chapter appears in a paper [82] jointly authored with Daniel Giger, Jordan Allspaw, Michael S. Lee, Dr. Henny Admoni, and Dr. Holly Yanco. Please see Publication 4.

from Demonstration [65] and navigation [107].

As we will discuss in Section 2.3, BTs have the advantages of modularity, reusability, and scalability, over more traditional state machines. Behavior nodes, the basic structure of BTs, are both modular and reusable, which allows the system to scale while maintaining fewer unique modules [40]. In contrast, states in state machines are tightly coupled and do not scale well to more complex tasks, due to intertwined transitions. This is especially relevant for situations in which different tasks are interconnected and involve multiple steps.

Some of the disadvantages of BTs include the fact that they are free-form and static. As a counterexample of being unconstrained, robot behaviors can still be specified in a very deep tree – making the resulting BT hard to justify and natively unsupportive for generating hierarchical robot explanations of shallow depth, similar to their human counterparts. And being static, BTs lack the ability of dynamic modification (e.g., existing behavior insertion). This disadvantage limits us from inserting an explanation represented by a self-contained partial tree.

To solve these problems, we frame BTs in semantic sets with a goal structure so that they are better suited for hierarchical robot explanation. We also propose explanation generation algorithms for the framed BTs, as well as an algorithm to create a self-contained behavior node that can then be dynamically inserted.

Additionally, we ground our framing structure and proposed algorithms with a multi-task, multi-step mobile manipulation kitting task. In this kitting task, a mobile manipulator robot assembles a kit of gearbox parts by navigating in a confined environment and collecting differently-shaped, irregular parts (Figure 19 in Section 4.3.1) from different stations. It also involves a peg-in-hole large gear insertion task for machining a large gear. By first modeling this kitting task, consisting of many different subtasks, using BTs, we show the strength in execution and richness of expression of BTs, in addition to its simplicity and friendliness for non-roboticists, as argued in [40] and shown by [134]. We also demonstrate the application of our work to a non-manipulation task, the taxi domain [53].

The primary contributions of this chapter are threefold:

1. We first model a complex kitting task in BTs to show robot developers that BTs are capable of specifying complex high-level robotic tasks while not losing simplicity.
2. Given that BTs can be deep, we frame BTs into semantic sets and contribute algorithms to generate hierarchical explanations, taking the node types in BTs into account.
3. We make the static BTs dynamic so self-contained behavior nodes can be inserted dynamically as a subgoal, as a member of the semantic set.

To validate the framing structure and proposed algorithms for explanation generation, we evaluate them on two diverse tasks, the first involving robot manipulation and the second involving navigation planning. The first task includes the large gear insertion subtask of the kitting task, in addition to the screw picking and placing subtasks that we used throughout this work to give readers a concrete idea about our work in the early phase. The second task involves a taxi routing problem that attempts to optimize an agent’s navigation path while completing subtasks like picking up and dropping off a passenger. By evaluating our explanation generation algorithms in these two different tasks, we demonstrate the algorithms’ generalization to domains with differing complexities and actions.

To the best of our knowledge, this is the first work on using behavior trees for explanation generation. We conclude with a discussion of this work, including its limitations and ideas for potential future work.

4.2 Background: Formulation of Behavior Trees

Behavior Trees have previously been used for gaming agents but have gained momentum in the robotics community during the last few years [110, 76, 39, 40, 135]. In this section, we borrow from the previous work to formulate Behavior Trees using a simplified yet complete set of notations.

A behavior tree $T = \{ C, E, Edges \}$ is an ordered rooted tree where *control flow nodes* $C = \{ Sequence, Fallback, Parallel, Decorator \}$ are internal or non-leaf nodes that have children nodes, and *execution nodes* $E = \{ Action, Condition \}$ are external or leaf nodes. Control flow nodes can have control flow or execution nodes as their children whereas execution nodes are execution units that do not have any children.

Execution starts or *ticks* from the root node of the tree. Control flow nodes route the ticks to its children until the execution nodes returns a status $s \in S = \{ RUNNING, SUCCESS, FAILURE \}$. The status of a control flow node depends on one of its descendant nodes.

A sequence control flow node, indicated by “ \rightarrow ” executes each of its child nodes sequentially. The sequence returns SUCCESS only if all of its child nodes return SUCCESS, and returns RUNNING or FAILURE if any of its children returns such status.

A fallback control flow node, denoted by “?” , also executes its children sequentially but only requires one child node to succeed in order to return SUCCESS. If a child node returns FAILURE, the fallback node will tick its next child until SUCCESS or FAILURE. If all of its children return FAILURE, the fallback node returns FAILURE. Similar to the sequence node, it returns RUNNING if any of its children returns such status. The fallback node type is also known as a selector.

A parallel control flow node, labeled as “ \Rightarrow ”, executes all of its children simultaneously or the children are all asynchronous themselves. Otherwise, the parallel node has the same control logic as a sequence node. Callers can specify thresholds to make the node return SUCCESS or FAILURE when a certain number of children nodes returns one of the statuses.

A decorator control flow node, often represented with a diamond shape, can only have one child. It applies a function f to its child to manipulate the returned status. Examples include an *inverter* node that negates its child’s status and a *retry* node that ticks its child N times as long as a FAILURE status is returned.

Usually boxed, an action node is an execution unit that runs a piece of code. It returns SUCCESS upon successful completion and FAILURE if it is impossible to complete. RUNNING

Table 4: Classical Behavior Tree (BT) Formulation: Nodes and Return Statuses. (Adapted from [40])

| Notation | Node | Node Type | $s = \mathbf{SUCCESS}$ | $s = \mathbf{FAILURE}$ | $s = \mathbf{RUNNING}$ |
|---------------|------------------|-----------|------------------------------|----------------------------------|----------------------------|
| \rightarrow | <i>Sequence</i> | Control | All children $\rightarrow s$ | Any child $\rightarrow s$ | Any child $\rightarrow s$ |
| ? | <i>Fallback</i> | | Any child $\rightarrow s$ | All $\rightarrow s$ | Any child $\rightarrow s$ |
| \Rightarrow | <i>Parallel</i> | | M children $\rightarrow s$ | $N - M$ children $\rightarrow s$ | Any child $\rightarrow s$ |
| Diamond | <i>Decorator</i> | | Function $f \rightarrow s$ | Function $f \rightarrow s$ | Function $f \rightarrow s$ |
| Boxed | <i>Action</i> | Execution | Self $\rightarrow s$ | Self $\rightarrow s$ | Ticking |
| Circled | <i>Condition</i> | | Self $\rightarrow s$ | Self $\rightarrow s$ | - |

is returned during execution.

Often circled, a condition execution node checks whether a state is met, making it a special case of the action node. However, it does not return the `RUNNING` state; otherwise it functions the same as an action node.

Table 4 lists all the nodes and their returned states.

4.3 Modeling Robotic Tasks using Behavior Trees

Even though the BT method is relatively simple, which makes it easy to understand for non-roboticists as previously mentioned [134], it does not mean that BTs are incapable of representing complex tasks or to execute those tasks. In order to demonstrate the execution model and the expressiveness of BTs, we now describe a real-world scenario of a complex gearbox kitting task and how one can represent tasks in BTs. We will refer this task when we describe how we use BTs for robot explanation.

4.3.1 The Gearbox Kitting Task

The gearbox kitting task is a challenging task designed for the 2019 FetchIt! Mobile Manipulation Challenge¹⁶, organized by Fetch Robotics and held at the IEEE Conference on Robotics and

¹⁶<https://opensource.fetchrobotics.com/competition>

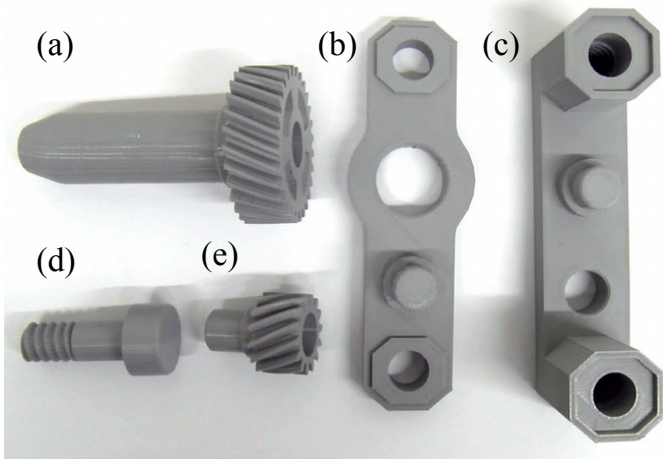


Figure 19: Parts to be collected: (a) Large gear (b) Gearbox top (c) Gearbox bottom (d) Screw (e) Small gear. Note that the large gear (a) is meant to be machined to have threads; the large gear must be inserted into a machine for this process to occur.

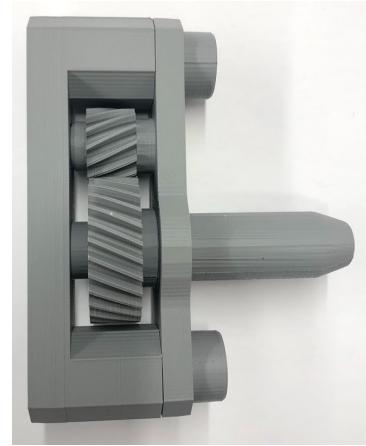


Figure 20: Assembled gearbox using the required mechanical parts that the robot placed into the caddy and delivered to the inspection table.

Automation (ICRA) in May 2019. In this competition, a Fetch mobile manipulator robot [170], which has a single chest-mounted 7 DoF arm with a torso lift joint and a head-mounted RGBD camera, needed to be programmed to assemble as many kits as possible. Kit assembly is achieved by navigating to different stations to pick a specified number of complex, irregular mechanical parts including screws, gears, and gearbox container parts (Figure 19), placing them into specific compartments of a caddy, and finally picking the full caddy up and delivering it to the inspection table for final gearbox assembly by factory or warehouse workers. The robot needs to first collect two screws from a bin as well as one each of a large gear, small gear, gearbox bottom, and gearbox top piece. We described our efforts to address the challenges present in this task in [79].

While the robot can complete the kitting task alone, the ultimate goal is to apply it to a collaborative scenario where one or two workers stay at the inspection table to inspect and perform the fine motor skills to assemble a functional gearbox (Figure 20) from the parts that the robot collected, requiring human-robot collaboration.

This human-robot collaboration scenario provides the tasks and the environment for a rea-

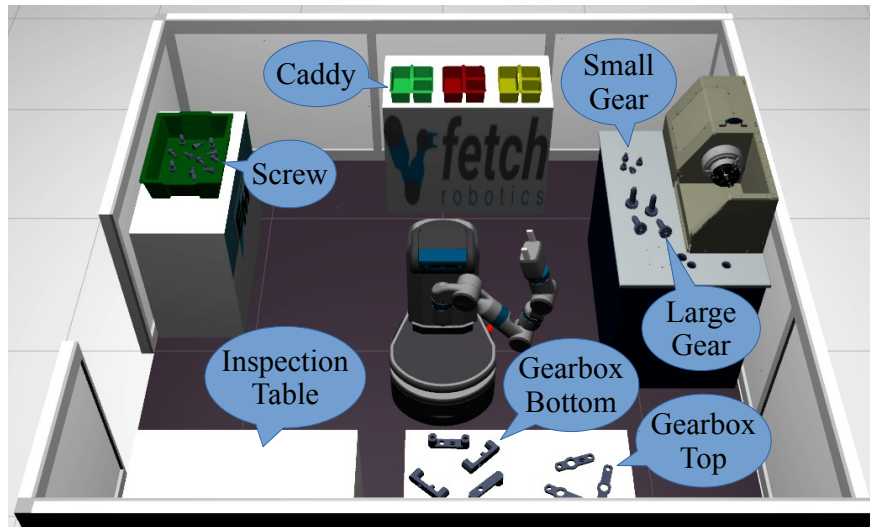


Figure 21: The arena where the gearbox kitting task is carried out by a Fetch robot. Rendered in Gazebo, the main goal is to place a specified set of parts into the correct sections of the caddy, then to transport the caddy to the inspection table.

sonable testbed. Because there are several opportunities for unexpected or opaque events to happen, where a human may want to ask for a robot to explain itself.

For example, a common occurrence is that the Fetch robot may not be able to grasp a caddy or a gearbox part if it is placed too close to a wall. Fetch’s arm may not be long enough to reach given the constraints presented by the end-effector orientation (it must be pointed down in order to grasp the caddy) and standoff distances imposed by the dimensions of the tables. This scenario is not apparent to novice users or bystanders who do not have intimate knowledge of Fetch’s characteristics. Even for roboticists, forming an inaccurate mental model of the robot from time to time is still possible. In such scenarios, Fetch should initiate an explanation to inform the user.

Another common occurrence is confusion when differentiating between two gearbox parts that appear similar in height via point cloud due to sensor noise. This can make the object detection fail, causing the robot to grasp the incorrect object. In this scenario, a human might initiate a robot explanation, as the robot may not be aware that it performed incorrectly.

Finally, a human-initiated robot explanation might also be needed when the robot stops at

a different location in front of the caddy table than what was expected, and places a part into the incorrect caddy. This could occur due to navigation error range and the narrow horizontal field of view (54°) of its RGBD camera, which may cause part of the caddy to be occluded.

While scoped for a manufacturing environment, the same types of tasks are relevant to home environments (e.g., navigating between areas in a narrow hallway kitchen and a dining table, and manipulating objects in these places).

In this paper, we focus on human-initiated high-level robot explanation.

4.3.2 Revised Notation for Behavior Trees

To visualize Behavior Trees, we adopted the Groot software¹⁷ and its notations. In contrast to the classical BTs notations summarized in Table 4, a subtree node is added. Every node is boxed to accommodate more text descriptions instead of a single word.

Text color is used to differentiate node type and custom decorator nodes on a dark gray background. For control nodes except for decorators, pink and blue indicate sequence and fallback respectively. For execution nodes, white and green indicate action (A) and condition (C) nodes. Because a decorator node is attached with a custom function f , they have different colors.

The subtree node can only have one child, similar to a decorator node, and is denoted by a tree hierarchy icon. Indeed, it can be deemed as a special case of a decorator with the function f simply doing nothing but ticking its child.

Table 5 shows the revised notations including the new subtree node and the color codes.

4.3.3 Modeling Using Behavior Trees

Figure 22 shows the top level of the gearbox kitting task represented as a BT where each subtask is encapsulated in a subtree. To improve the readability of the tree in this paper, all subtasks were left collapsed in the figure.

¹⁷<https://github.com/BehaviorTree/Groot/>

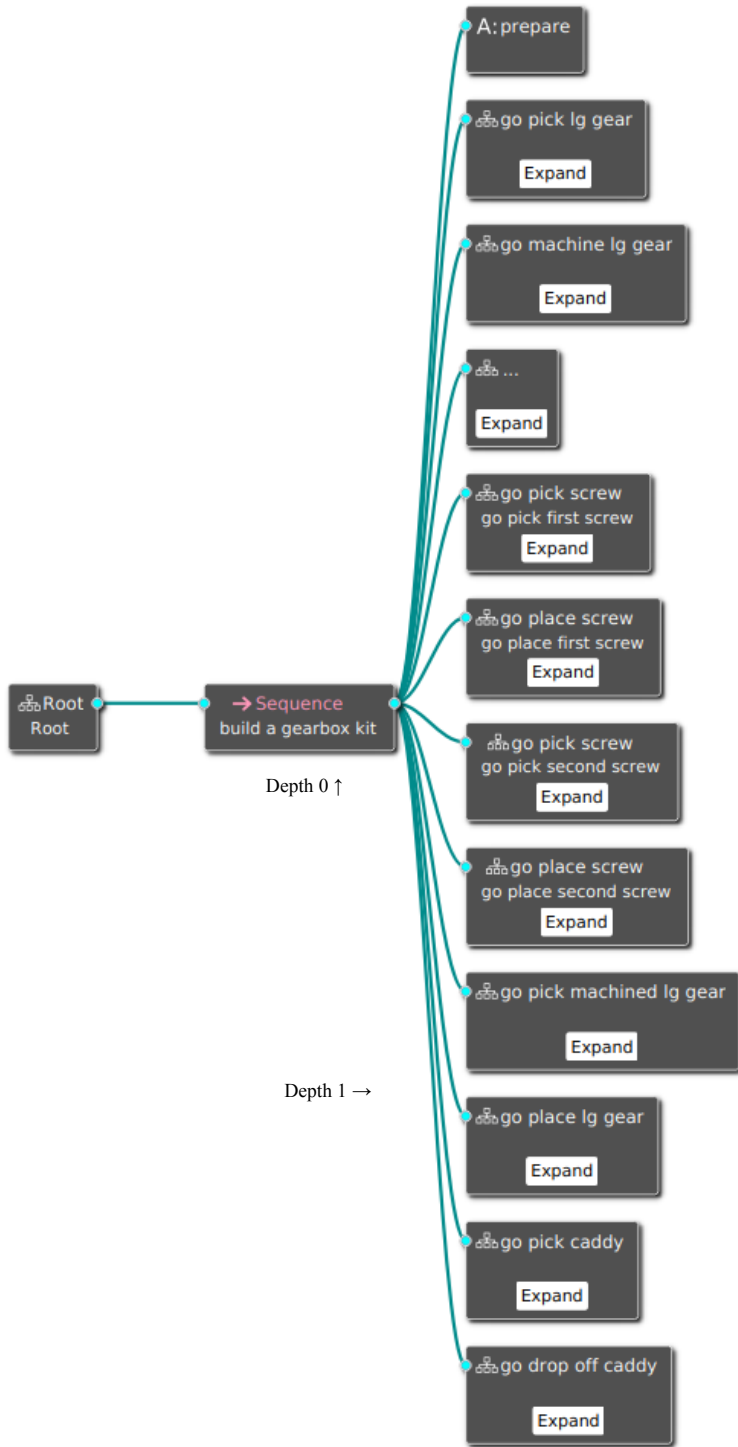


Figure 22: The top level of the gearbox kitting task represented as a sequence of subtrees in a Behavior Tree. For readability, six tasks – go {pick — place} {small gear — gearbox top — gearbox bottom} – are represented by “...”. Note that the root node on the top is merely a pointer to the real root node in the middle, which is why depth 0 is at the sequence node.

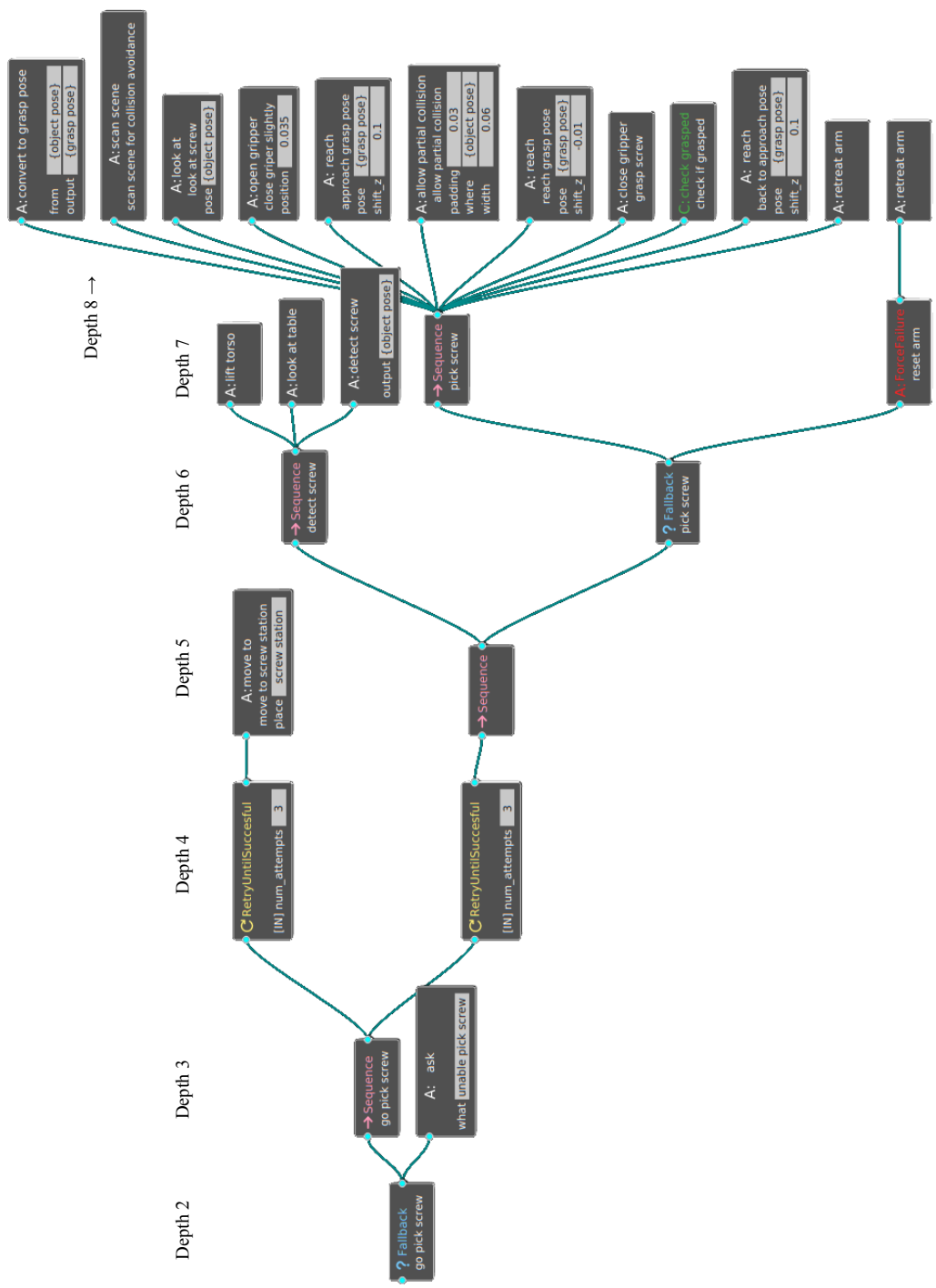


Figure 23: The representative screw picking subtask modeled in Behavior Trees. The leftmost dot indicates that the fallback node has a parent, but the subtree parent node and other ancestor nodes are hidden here as they are shown in the previous figure.

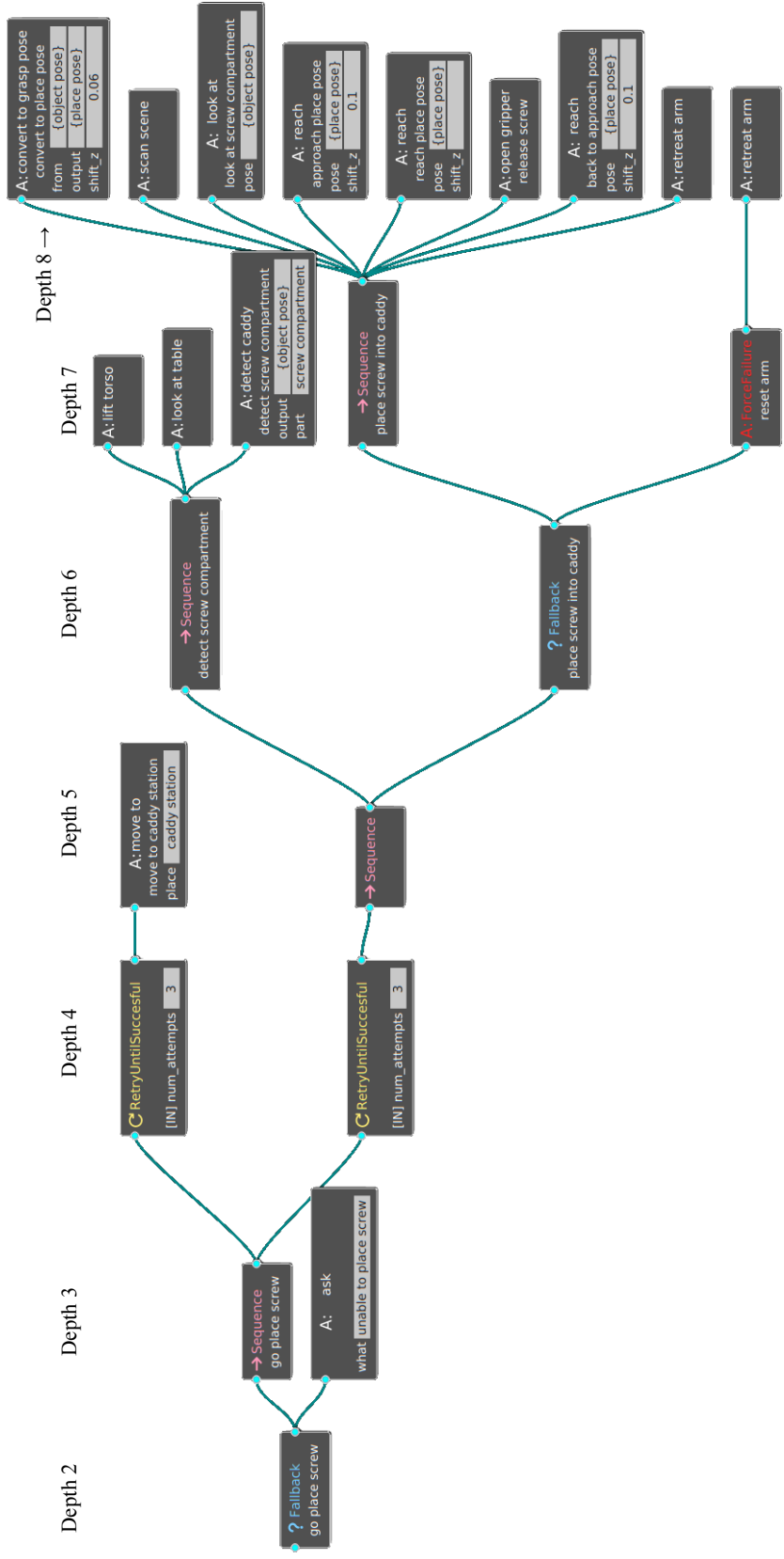


Figure 24: Screw placing, another representative subtask of the gearbox kitting task, modeled as a Behavior Tree. Similar to the previous figure, the leftmost node's ancestors are not shown. For more detail, refer to Section 4.4.

Table 5: Notation of and changes to Behavior Trees as used in this chapter

| Symbol | Color | Node | $s = \text{RUNNING}$ |
|---------------|--------|----------------------|------------------------------|
| | White | <i>Subtree</i> | Child $\rightarrow s$ |
| \rightarrow | Pink | <i>Sequence</i> | Any child $\rightarrow s$ |
| ? | Blue | <i>Fallback</i> | Any child $\rightarrow s$ |
| | Varies | <i>Decorator</i> | Function $f \rightarrow s$ |
| | White | <i>Action (A)</i> | Ticking |
| | Green | <i>Condition (C)</i> | Ticking |

Figures 23 and 24 illustrate the specification of two representative subtasks in the kitting task using BTs: picking a screw placed in a bin on the table and placing a screw into a specific caddy compartment (See Figure 21 left and top). Other tasks in the kitting task are similar and can be represented in the same way, thus highlighting the modularity and reusability of BTs by reusing existing tree nodes. For example, picking gearbox tops/bottoms and large/small gears are similar to screw picking. Placing those parts into a caddy is the same as placing a screw into the caddy except that the compartment is different. If you compare Figures 23 and 24, more than half of the nodes are reused. However, machining a large gear is a unique task. We visualize it in Figure 27 and will discuss in Section 4.7 during evaluation in hope of generalization. The examples demonstrate every type of node in both the control and execution categories except for the parallel node. “RetryUntilSuccessful” and “ForceFailure” are two types of decorator nodes.

As illustrated in Figure 23 and 24 as well as Figure 27 to be seen later, each node is visualized with its node ID on the first line, node name on the second line, and input and output (I/O) ports following. Custom nodes can register their own IDs and are mostly at the leaf level (i.e., action and condition nodes). Node names are not needed for registration and only used when the node is reused to clarify the robot behavior it represents.

With the built-in nodes listed in Table 5 and registered custom nodes, the whole tree, including descriptive names and I/O ports, is specified in an XML file which is then parsed for execution or visualization. As an example, the bottom right action node in Figure 23, `retreat arm`, has

the ID of “retreat arm” and an empty name because the ID shows the specific behavior that the node represents. At the same level, the “reach” node is reused 3 times and assigned a name each time to differentiate each other. As we will see later during explanation generation, the node name plays an important role as it can represent the robot behavior.

Input and output ports are the light gray fields filled with string values. Similar to the previously mentioned SMACH [23], these ports are designed to be data-driven and passed around between nodes to share data. Input ports can be specified by a literal value or a dynamic key, while output ports, named “output”, can only be keyed. For example, the top right action node in Figure 23, “convert to grasp pose”, has a “from” input port with a “{object pose}” key and a output port with a “{grasp pose}” key. Each tree or subtree maintains a key-value dictionary to store outputs resulted from some nodes and to be used by other nodes in the tree or subtree. If subtree nodes are used, each dictionary is scoped within the subtree to avoid naming space collision for the keys and be easy to reason about.

4.3.4 The Screw Picking Subtask: Tree Breakdown

As visualized in Figure 23, the `go pick screw` task starts as a fallback node (depth 2), which ticks the sequence node with the same name. If the sequence node fails, the `ask` node will be executed to ask humans to intervene. In the children of the sequence node (depth 4 and 5), a robot will first `move to screw station` and, if unsuccessful, retry three times. Then the robot will `detect screw` and `pick screw`, and retry up to three times until successful. A sequence node with an empty name (depth 5) is needed as the `RetryUntilSuccessful` node is a decorator node which can only have one child. To `detect screw`, the robot needs to `lift torso`, `look at table`, and `detect screw` which outputs `object pose` of the screw. A fallback node is used to `pick screw` (depth 6) so, when the sequence of `pick screw` failed, the robot will `reset arm` (depth 7 bottom) before retrying (the second child at depth 4). To `pick screw`, the robot will run all execution nodes at depth 8. For the action node of `allow`

partial collision, it will crop a 4mm cube at `object pose` with 3mm padding from the collision scene, allowing the gripper to collide with the detected screw.

The `go place screw` task in Figure 24 can be broken down in a similar fashion. Note that this screw placing task shared a considerable amount of sequence and action nodes, illustrating the modularity and reusability advantages of using BTs again. Thanks to this, once the first behavior tree is coded, similar tasks can be easily encoded in behavior trees using existing nodes.

4.4 Framing Behavior Trees for Hierarchical Explanation Generation

Behavior Trees are a free-form action sequence specification and execution tool that only provides a preset of control nodes with simple control flow algorithms, imposing a minimum amount of structural rules. This structure provides ultimate flexibility for robot or AI developers to program behaviors for different application needs or use cases, such as the manipulation tasks in this paper.

However, to provide concise, multi-level explanations [80], we propose to frame BTs for hierarchical explanation generation, given the experience from implementing the subtasks in the kitting task with BTs.

The behavior tree of a multi-task, multi-step task can be simplified and decomposed into a set of semantic sets:

$$\{goal, subgoals, steps, actions\}.$$

To give a concrete understanding, now we examine the kitting task tree. As shown in Figure 22, the *goal* is to `build a gearbox kit` and its subgoals are all of the leaf nodes. A *subgoal* can be further broken down into *steps*, which can be nested to span more than one level or as shallow as a flat one-level tree. Finally, the ending steps consist of a set of *actions* (i.e., execution nodes).

Figure 25 and 26 shows the framed behavior trees of the screw picking and placing subtasks. The framed tree can be generated by ignoring decorator nodes and the control nodes that have a

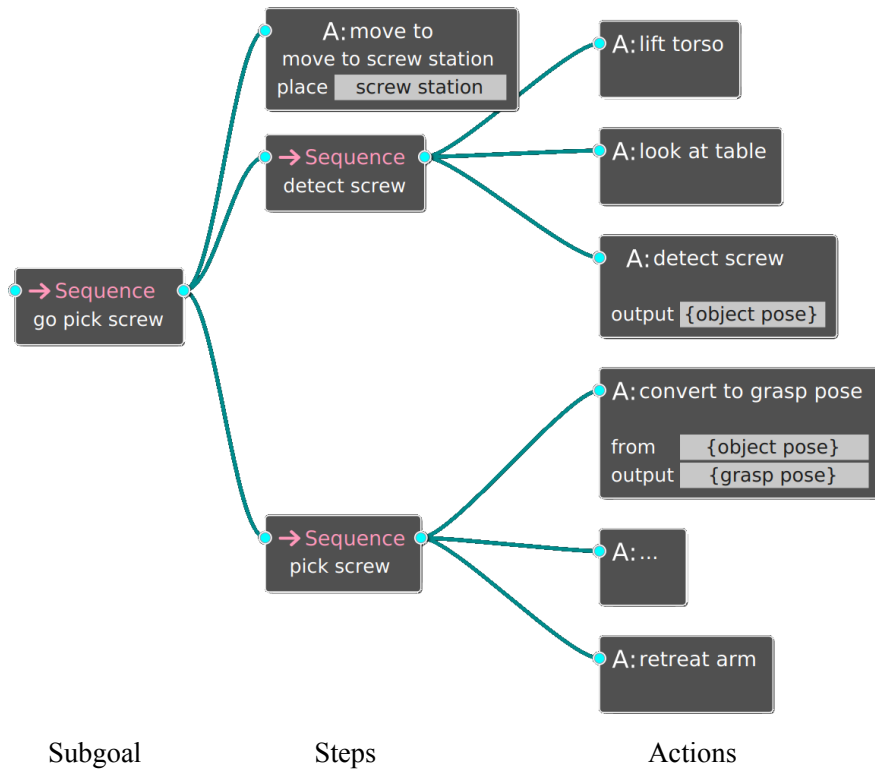


Figure 25: Simplified, semantic sets for the screw picking subtask. The goal is not shown here as Figure 22 is sufficient without any simplification.

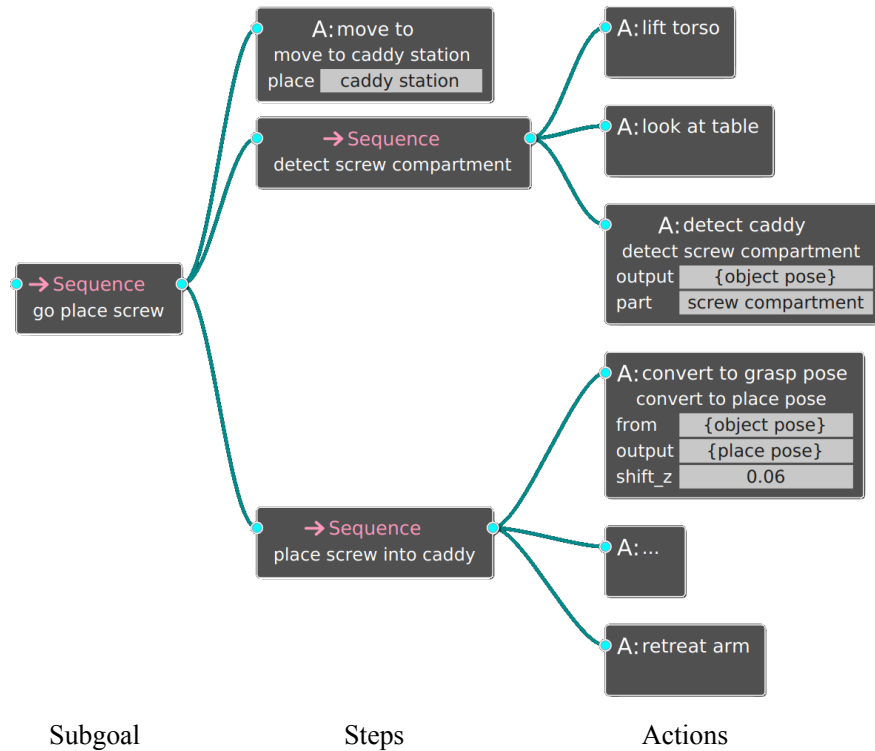


Figure 26: Simplified, semantic sets for the screw placing subtask.

child sharing the same name. For example, all `RetryUntilSuccessful` decorator nodes are ignored, and the fallback nodes at depth 2 and 6 are also ignored (see Figure 23). Note that the `move to` node is a special case of a step, which reveals that a step does not have to be a control node.

Compared to the original behavior trees in Figure 23 and 24, the hierarchy is much simplified and more clear, making it more suitable for explanation generation. Section 4.5.2 details the generation algorithms that used the framed trees under the hood.

4.5 Algorithms on Behavior Trees for Robot Explanation Generation

Behavior Trees are a static action sequence method. Once created, it does not allow for the dynamic addition of behavior nodes. BTs are also not designed to be interactive. The behavior tree associated with a task might be presented on a display screen for introspection, but when an end-

Algorithm 1: Answer “What are you doing?” (Q1)

Input: Node n (Current node in execution)**Output:** String $answer$

```
1  $p.short\_description \leftarrow$  if  $p.has\_name$  then  $p.name$  else  $p.ID$  end;  
2 return "I " +  $p.short\_description$  + ".";
```

user asks what part of the task the robot is attempting to finish or why the robot is doing its current sub-task, it is unknown how to query at different nodes or what should the robot reply given different functions of different types of node. To solve this issue, we propose the following algorithms. All the pseudo-code is derived from our C++ implementation using the `BehaviorTree.CPP` library¹⁸. The C++ implementation is available on GitHub¹⁹.

4.5.1 Supporting Querying Current State

Because any BT node returns a status $s \in S = \{ \text{RUNNING}, \text{SUCCESS}, \text{FAILURE} \}$ and condition nodes do not return a RUNNING state, we reintroduce the RUNNING state back to condition nodes in order to know which node is currently running. This change is reflected in Table 5.

This change is necessary because robots operate in the real three-dimensional world in which completing a checking task is more time-consuming than checking a condition in a virtual world. Behavior Trees are popular in the gaming industry, in which inputs are accurate, checks are fast, and noisy sensors do not exist. On the other hand, it may take time for an embodied robot to check if a state is reached (i.e., execute the condition node). For example, to check if a screw in a bin is reachable, it can take a while for sampling-based motion planning methods to avoid collisions for small object manipulation, especially when the collision objects are further away and the objects are represented as a number of small units, making the scene representation time-consuming to search.

Given that every execution node can be in the state of RUNNING, we can use the concept of

¹⁸<https://www.behaviortree.dev/>

¹⁹<https://github.com/uml-robotics/robot-explanation-BTs>

Table 6: Questions To Be Answered During Hierarchical Explanation

| Question | Algorithm |
|--|----------------|
| Q2 Why are you doing this? | Algorithm 2 |
| Q3 What is your subgoal? | Algorithm 3 |
| Q4 What is your goal? | Algorithm 4 |
| Q5 How do you achieve your { goal — subgoal }? | Algorithm 5, 6 |

a state listener to track which node is ticking. The tracked node with the RUNNING state can then be used to answer the most basic question: Q1. “What are you doing?” For the sake of showing all of the algorithms in this text, we include this as Algorithm 1.

4.5.2 Supporting Hierarchical Explanation Generation

Once we have the framed Behavior Trees for hierarchical robot explanation, the context to answer the questions listed in Table 6 is established by the sets of semantic sets, in addition to Q1.

Compared to Q1, these questions are chosen because humans tend to seek causal knowledge from explanations [105, 22]. Q2 gives the causal information of Q1. Q3 gives the intermediate cause while Q4 offers the final cause – “the end, function or goal” [105] and Q5 is designed in the hope of providing detailed steps to improve understanding. Q1–Q4 are about thinking backward in term of the tree structure while Q5 is forward-thinking, from left to right.

Note that we do not focus on speech recognition or natural language processing (NLP). With the underlying algorithms described in this section, one could plug in an NLP framework to extract the semantic information from the question. In our work, we are also not concerned with the grammatical correctness of the explanations; we could also add additional language generation capabilities.

Algorithm 2 shows the steps to answer Q2, reasoning about the current behavior. Lines 1–5 find the non-decorator control ancestor node p of the node in execution n so p has a name and the name is different from n 's. Note that the short description of a node is described in Algorithm 1: It is the name of the node n' if n' has a name, otherwise the ID of n' .

Algorithm 2: Answer “Why are you doing this?” (Q2)

Input: Node n (Current node in execution)**Output:** String $answer$

```
1  $p \leftarrow n.parent$ ;  
2 // find a non-decorator ancestor of “ $n$ ” that has a name different from “ $n$ ”;  
3 while  $p \neq null$  or  $p.type = Decorator$  or  $p.has\_name()$  or  $p.short\_description =$   
    $n.short\_description$  do  
4   |  $p \leftarrow p.parent$ ;  
5 end  
6 return “I ” +  $p.short\_description$  + ” in order to ” +  $p.name$  + ”.”;
```

Algorithm 3: Answer “What is your subgoal?” (Q3)

Input: Node n (Current node in execution)**Output:** String $answer$

```
1  $p \leftarrow n.parent$ ;  
2 while  $p \neq null$  and  $p.type \neq Subtree$  do  
3   |  $p \leftarrow p.parent$ ;  
4 end  
5 if  $p \neq null$  then return “My subgoal is to ” +  $p.name$  + ”.”;  
6 else return “Sorry. I don’t have a subgoal.”;
```

Algorithm 3 shows the steps to answer Q3 about subgoal. Lines 1–4 find the subtree ancestor node of n as the subgoal.

To answer Q4, the goal is simply the name of the root node, as indicated by Algorithm 4.

To answer Q5, Algorithm 5 shows the steps where a depth-first search is performed at the goal or subgoal node g . In lines 3–6, a descendant node is added to the set of steps if the node being traversed is not a decorator node, has a name, and does not have the same name as g . Similar to previous algorithms, Algorithm 6 shows the generated explanation text.

4.5.3 Supporting Failure Explanation Generation

In addition to asking for explanations for the current behavior, explanations from failure handling are of interest. For example, “Was there anything wrong?”, “What went wrong?”, and “How was the failure handled?”.

Algorithm 4: Answer “What is your goal?” (Q4)

Input: Node *root***Output:** String *answer*

```
1 return "My overall goal is to " + root.name + " .";
```

Algorithm 5: Find Steps From a Goal or Subgoal Node

Input: Node *g* (The goal or subgoal node)**Output:** Set *steps* (An ordered set of nodes)

```
1 steps ← ∅
2 DepthFirstSearch: g, f(node) → begin
3   | if node.has_name() and node.name ≠ p.name and node.type ≠ Decorator then
4   |   | steps ← steps ∪ {node};
5   |   | do not traverse further;
6   |   | end
7   | end
8 return steps;
```

All those questions can be answered by Algorithm 7, which centers around Fallback and RetryUntilSuccessful nodes that handle failures. For a Fallback node, when its status is “SUCCESS”, we can check whether the first children nodes in its sequence were unsuccessful (line 4), i.e., to see whether the execution fell back to other child nodes. Those children nodes with “FAILURE” status (lines 7–9) can then be used for explanations of failures, more specifically, what went wrong (lines 6 and 10–12). For example, in the Fallback node at depth 6 of Fig. 23, if its “pick screw” child node failed because the “check if grasped” failed, the robot can say “I could not go pick screw because check if grasped failed.” Potentially, it can be further explained by attaching information such as perception results and motion trajectories; the latter two can be presented on a monitor or by a projector.

For a RetryUntilSuccessful decorator node, it can also be used to answer the questions, especially that the failure is being handled by retries. The relevant code in Algorithm 7 are lines 13–33. Line 13–23 handles the case when the Retry node has started its first attempt (line 13), indicating one node has failed. Then we can find its parent to explain what is being attempted

Algorithm 6: Answer “How do you achieve your { goal — subgoal } ?” (Q5)

Input: Node n (Current node in execution)

Input: Node g (The goal or subgoal node found in Algorithm 3)

Output: String $answer$

```
1 if  $g = null$  then // specific to subgoal
2   | return "Sorry. I don't have a subgoal."
3 else
4   | steps  $\leftarrow$  output of Algorithm 5 given  $g$ ;
5   | return "To achieve the {goal — subgoal} " +  $g.name()$  + ", I need to " +
6     | steps.to_string();
7 end
```

(lines 15–17). Similar to the Fallback node, the failed node is then found to explain what went wrong (line 18–22). Line 25–33 considers the case where a Fallback node has an ancestor Retry node. In this case, we want to inform the information at the retry node (which attempt and what it is attempting to do) in addition to the Fallback information in the previous paragraph. Note that the algorithm does not handle the case that a Retry node has an ancestor Fallback node. From our experience, we found that Fallback is a better choice to handle failure than Retry, because there might be another preferred method over retrying.

4.5.4 Supporting Dynamic Behavior Insertion as Subgoal

Behavior Trees have the advantages of modularity and reusability, but we cannot simply take any node n' and insert n' after the current execution node n , because some behavior nodes have dynamic input ports and are thus dependent on some other behavior nodes providing corresponding output ports. For example, if a user asked the robot to `place screw` (i.e., the sequence node at depth 7 of Figure 24), we cannot simply insert the sequence node because its children need `object pose` as input, which is provided by the `detect caddy` node at depth 7. We thus need a way to not only directly insert the behavior node being asked, but finds a self-contained node whose descendants provide the corresponding output ports.

Algorithm 8 shows the steps to find a self-contained behavior node n_s . First, we find the

Algorithm 7: Answer “Was there anything wrong?” “What went wrong?” “How was the failure handled?”

Input: Node n (Current node in execution)
Output: String $answer$

```

1  $p \leftarrow n.parent$ ;
2  $is\_wrong, fell\_back \leftarrow false$ ;
3 while  $p \neq null$  and  $p.type \neq Subtree$  and  $is\_wrong \neq true$  do
4   if  $p.type = Fallback$  and  $p.children[0].failed$  then // Fallback node
5      $is\_wrong, fell\_back \leftarrow true$ ;
6      $answer \leftarrow "I\ could\ not\ " + p.short\_description() + " because "$ ;
7      $n_{fail} \leftarrow DepthFirstSearch: p, f(node) \rightarrow begin$ 
8       if  $node.failed$  and  $node.type \in \{ Condition, Action \}$  then return  $node$  ;
9     end
10    if  $n_{fail}.parent \neq null$  and  $n_{fail}.parent.short\_description = p.short\_description$ 
11      then
12         $answer += "I\ was\ unable\ to\ " + n_{fail}.parent.short\_description + " as "$ ;
13         $answer += n_{fail}.short\_description + " failed."$ ;
14    else if  $p.type = Retry$  and  $p.attempt \leq 0$  then// Retry node
15       $is\_wrong \leftarrow true$ ;
16       $rp = p.find\_non\_null\_parent$ ;
17      if  $rp \neq null$  then
18         $answer = "I\ am\ retrying\ for\ attempt\ " + p.attempt + " to "$  +  $rp.short\_desc +$ 
19           $"."$ ;
20         $n_{fail} \leftarrow DepthFirstSearch: p, f(node) \rightarrow begin$ 
21          if  $node.failed$  and  $node.type \in \{ Condition, Action \}$  then return
22             $node$  ;
23          end
24           $fp = n_{fail}.first\_ancestor\_with\_name$ ;
25           $answer += "I\ could\ not\ " + fp.short\_desc + " because "$  +  $n_{fail}.short\_desc$ 
26             $+ " failed."$ 
27         $p \leftarrow p.parent$ 
28      end
29    if  $fell\_back$  then // Check if the Fallback node has Retry parent
30       $p \leftarrow fallback\_node.parent$ ;
31      while  $p \neq null$  and  $p.type \neq Subtree$  do
32        if  $p.type = Retry$  and  $p.attempt \leq 0$  then
33           $rp = p.find\_non\_null\_parent$ ;
34          if  $rp \neq null$  then
35             $a += " I\ am\ retrying\ for\ attempt\ " + p.attempt + " to "$  +  $rp.short\_desc +$ 
36               $"."$ 
37             $p = p.getparent()$ ;
38          end
39        end
40      if  $not(is\_wrong)$  then  $answer = "Nothing\ went\ wrong."$  ;
41    return  $answer$ ;

```

Algorithm 8: Find Self-Contained Behavior Node

Input: Node n (The node matching what is requested and $n.type \in \{ Sequence, Subtree \}$)

Output: Node n_s (Self-contained execution or control node where keyed input parameters of its descendants are outputted from its descendants)

```
1 // get all input ports possibly with duplicates;
2 input_ports  $\leftarrow \emptyset$ ;
3 foreach  $e \in n.execution\_descendants$  do
4   |  $input\_ports \leftarrow input\_ports \cup e.input\_ports$ ;
5 end

6 // filter all dynamic, keyed input ports without duplicates;
7 dynamic_input_ports  $\leftarrow$  select distinct * from input_ports where  $_.is\_keyed()$ ;

8 // find an ancestor node who or whose descendants output(s) all dynamic input ports;
9  $n_s \leftarrow n$ ;
10 foreach  $p \in dynamic\_input\_ports$  do
11   | if  $n_s.execution\_descendants.has\_output\_port(p.type, p.key)$  then
12     | continue;
13   | else // go up a level
14     |  $n_s \leftarrow n_s.parent$ ;
15     | while  $not(n_s.execution\_descendants.has\_output\_port(p.type, p.key))$  do
16       |  $n_s \leftarrow n_s.parent$ ;
17     | end
18   | end
19 end
20 return  $n_s$ 
```

shallowest parent n of the action as the input of the algorithm. From lines 2–7, we get a unique set of the dynamic input ports from each execution descendant of n . From lines 9–19, we try to find all the output ports that provide data to the dynamic ports from n and, in the `else` block from lines 14–17, if n does not provide any dynamic input port, we traverse the ancestors of n .

Algorithm 9: Append Self-Contained Behavior Node As a Subgoal

Input: Node n (Current node in execution)
Input: Node n_s (Self-contained node, output from Algorithm 8)
Input: Node $root$

```

1 // find the Subtree parent of “n” as current subgoal;
2  $p \leftarrow n.parent$ ;
3 while  $p \neq null$  and  $p.type \neq Subtree$  do
4   |  $p \leftarrow p.parent$ ;
5 end
6 assert( $p \neq null$ , “must use Subtree as subgoal”);
7 assert( $p \neq root$ , “must have a Subtree as subgoal”);
8 // insert “ $n_s$ ” after the current subgoal;
9  $root.insert\_child\_after(n_s, p)$ ;

```

Algorithm 9 shows the steps to insert the self-contained node n_s after the current subgoal. Lines 2–5 find the current subgoal given the current node in execution n . Line 9 inserts n_s as a subgoal.

Before we move onto the next section, we have an important implementation note for practitioners. As the BTs are static, existing implementations may get the number of children before executing any child to check if every child is ticked. However, because we add the ability to dynamically insert a child, the number of children can be changed while a child is executing. During our implementation of Algorithm 8 and 9, we used a function to dynamically find the next sibling by locating the current running child and checking out of bound every time.

4.6 Case Studies

In this section, we explore additional use cases of the algorithms presented in this paper, including a gear insertion task for machining and explanations in the taxi domain [53]. The two domains cover many applications in robotics, from manipulation to navigation, in continuous as well as discrete space.

4.6.1 Large Gear Insertion: A Machining Task

To evaluate the algorithms, we first apply them to the a large gear insertion task, one that would be found in a manufacturing application. Note that we have also demonstrated the proposed algorithms on the screw picking and screw placing tasks, which we described earlier as examples to give readers a concrete idea next to the abstract discussions. As mentioned earlier, unlike the screw picking and placing tasks, the insertion task is a different task as it is not a pick-and-place task but involves peg-in-hole insertion.

The behavior tree representation for the large gear insertion task is shown in Figure 27 and the framed tree is shown in Figure 28.

4.6.2 Hierarchical Explanation Generation

We first briefly review the algorithms to answer Q1–Q5. Answering Q1 is straightforward, which only needs to return the short description of the node in execution. Answering Q2 requires recursively visiting the ancestors of the current node n in execution, ignoring those without a name and those with the same name as n 's, as well as decorator nodes. Answering Q3 needs to find the subtree parent of n , which may not always exist. Answering Q4 simply returns with the name of the root node. Answering Q5 involves performing a depth-first search for the descendants of n until it finds a child that has a name that is not the same as n 's. During the process, it ignores all decorators and, when one of the children is found, the branch is not explored any further.

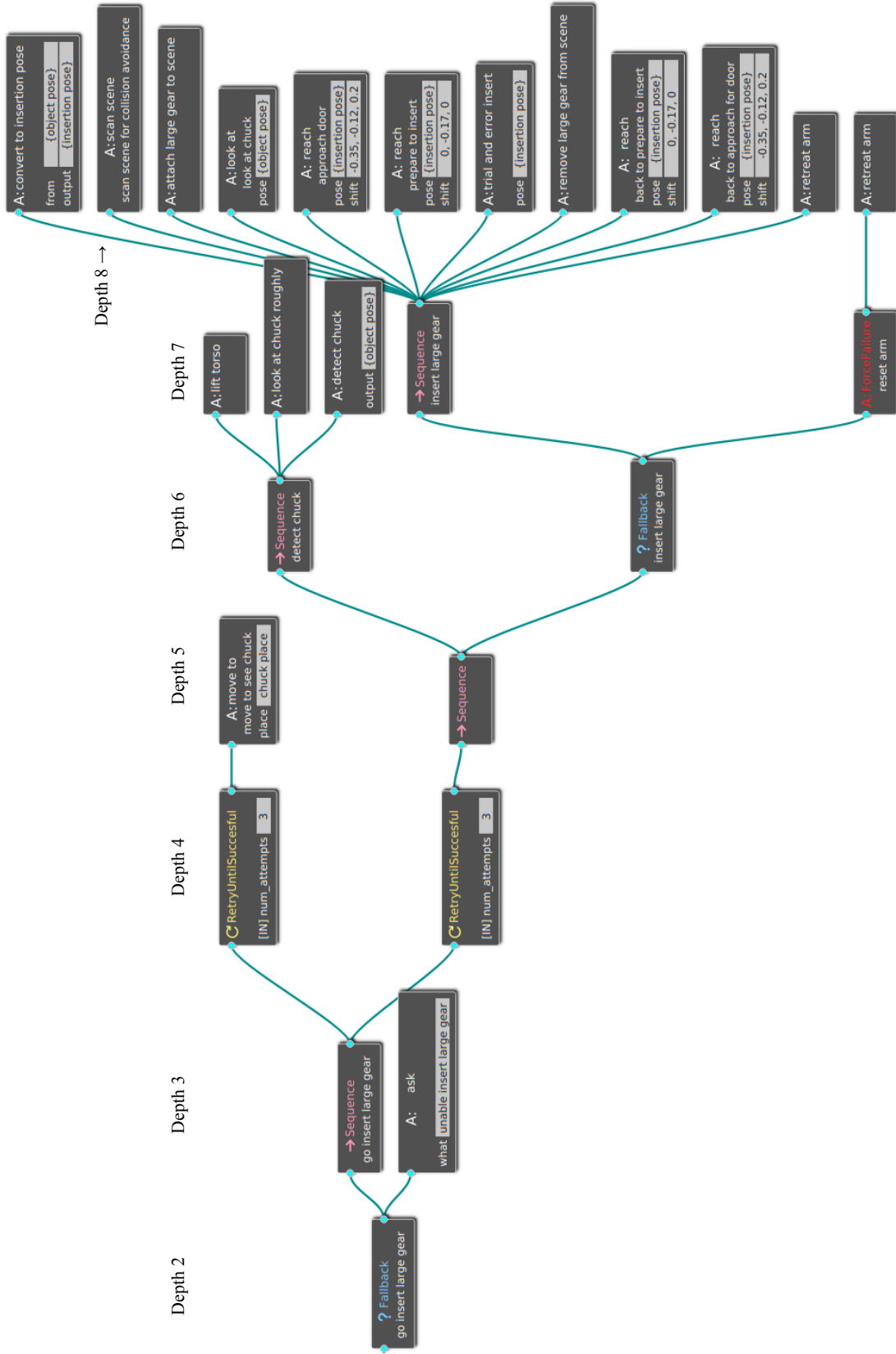


Figure 27: The large gear insertion subtask modeled in Behavior Trees.

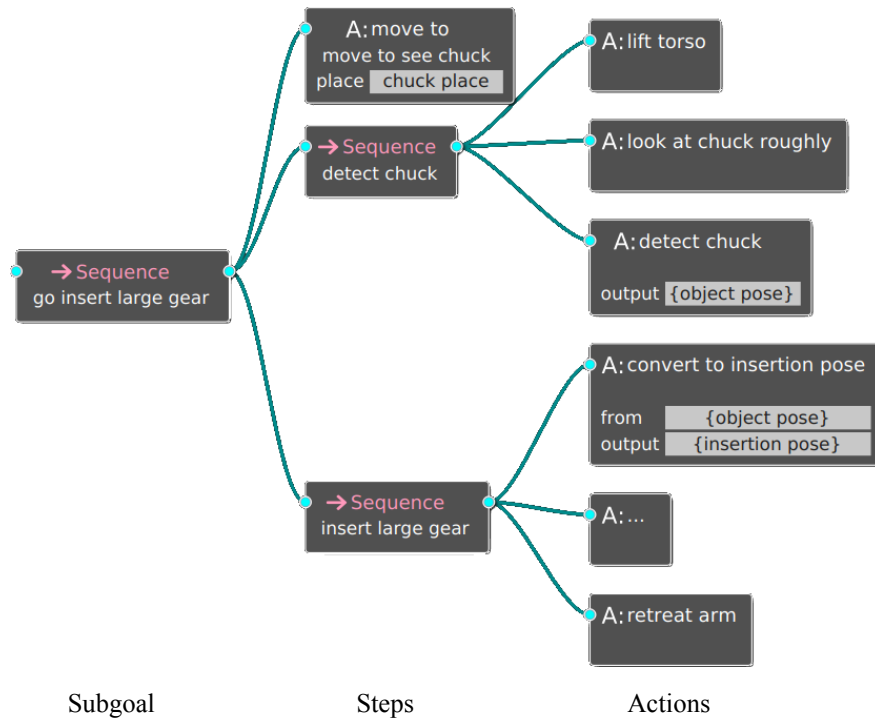


Figure 28: The framed BT for the large gear insertion subtask.

We now validate the generation algorithms by examining representative nodes in the large gear insertion task shown in Figure 27.

The simplest case is the `move to` node at depth 5 across all 3 subtasks. It has a `RetryUntilSuccessful` decorator parent and then a subgoal parent. In the context of the gear insertion subtask, the answers to Q1 and Q2 are “I move to see chuck” and “I move to see chuck in order to go insert large gear”. In answering Q2, the `RetryUntilSuccessful` decorator is ignored. The answers to Q3 and Q4 are “My goal is to build a gearbox kit” and “My subgoal is to go insert large gear”. When asked how the robot achieves the subgoal or goal, i.e., Q5, the answers are the following:

To achieve the subgoal “go insert large gear”, I need to do 3 steps. 1. move to see chuck. 2. detect chuck. 3. insert large gear.

To achieve the goal “build a gearbox kit”, I need to do 3 steps. 1. go insert large

gear. 2. go pick screw. 3. go place screw.

Note that the subgoal answer to Q5 is to verbally state the semantic set of “steps” shown in Figure 28.

A relatively deeper example in terms of tree depth are the `look at . . .` nodes at depth 7 across all 3 subtasks. It has an immediate sequence node parent which has a sequence parent without a name because the `RetryUntilSuccessful` decorator parent can only have one child. In the context of the gear insertion subtask, the answers to Q1 and Q2 are “I look at chuck roughly” and “I look at chuck roughly in order to detect chuck”. The answers to Q3–Q5 are the same because the goal and the subgoal are shared.

Another example is from one of the execution nodes at depth 8. Let’s take the `scan scene` node as the example. Still in the context of the gear insertion subtask, the answers to Q1 and Q2 are “I scan scene for collision avoidance” and “I scan scene for collision avoidance in order to insert large gear”. The answer to Q3 is “My subgoal is to go insert large gear”. The answers to Q4 and Q5 are the same as before.

With those three examples, one can easily infer the answers in the screw picking and place subtasks.

4.6.3 Dynamic Behavior Insertion as Subgoal

As we went through the insertion algorithm in the previous section, we will now consider how to answer the question “Can you insert large gear?” The `insert large gear` sequence node i at depth 7 will be first located. Then each of the decedents of i , from `convert to insertion pose to retreat arm` at depth 8 in Figure 27, will be examined to get a unique set of dynamic input ports: $\{object\ pose, insertion\ pose\}$. The *object pose* will be found from the `detect chuck` node (depth 7), a grandson of i ’s grandparent, the empty sequence node (depth 5). The *object pose* input port will be found right at `convert to insertion pose` which is a child

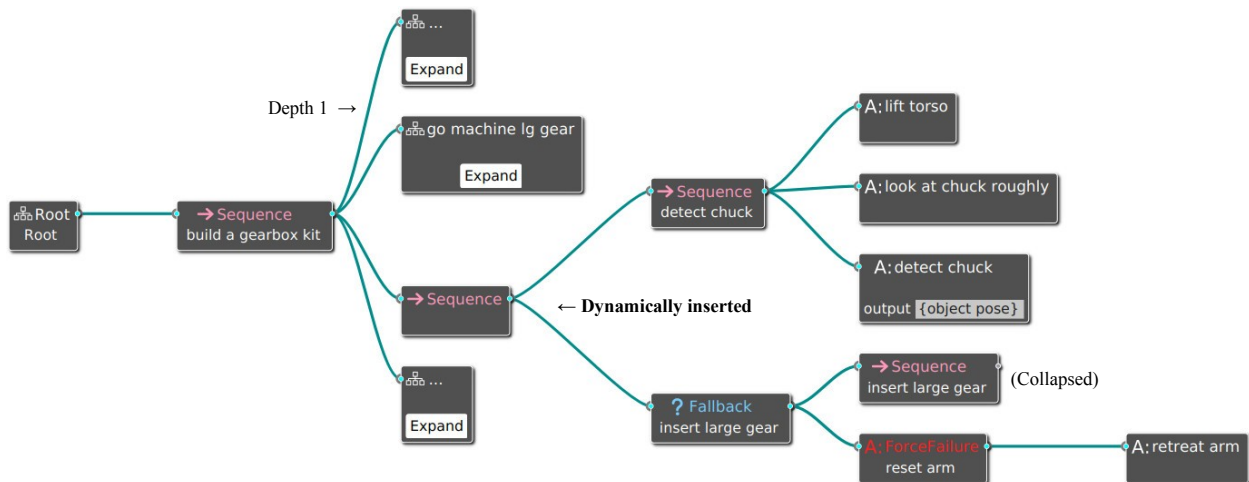


Figure 29: The behavior tree for the kitting task after answering ““Can you insert large gear?”. Most subgoals in Fig. 22 are collapsed for readability. The empty sequence node at depth 1 is dynamically inserted to ensure all input ports are satisfied by output ports. See Section 4.6.3 for more.

of i . Thus, the empty sequence at depth 5 will be inserted after the subgoal node at depth 1 (Figure 22). The final tree is shown in Fig. 29 with some nodes collapsed for readability.

The same logic also applies if a user asks “Can you place screw into caddy?” or “Can you pick screw?”

4.6.4 Explanations of Failures

Failures can happen at any node, so it is important to consider timing. Here we will test Algorithm 7 in different scenarios. Again, we will use the behavior tree in Fig. 27.

If the failure happened at the “look at chuck” node (fourth node at depth 8) and the question is asked while the robot is retreating its arm, the answer will be “I could not insert large gear because look at chuck failed.” The “insert large gear” is the description of the Fallback node (second node at depth 6). If the question is asked while “look at chuck” failed for the second time, it will answer “I am retrying for attempt 1 to go insert large gear. I could not insert large gear because look at chuck failed.” Here the retry information is added.

A failure can also happen during navigation, i.e., the “move to” node at depth 5. If the robot

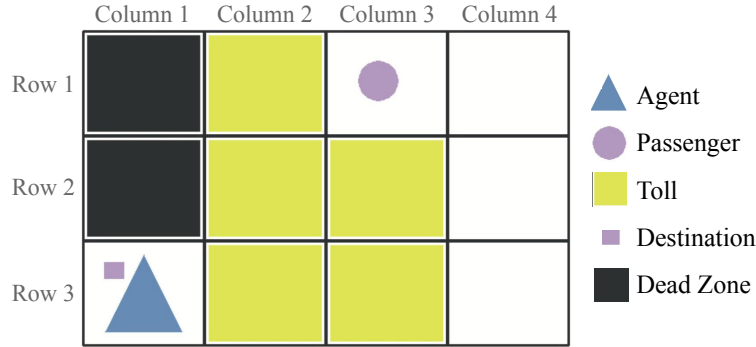


Figure 30: An environment of the taxi Domain, in which the agent delivers a passenger to the destination. Row and column numbers are added for easy reference.

keeps navigating for a long time and a person wonders why. The answer will be “I am retrying for attempt 1 to go insert large gear. I could not go insert large gear because move to see chuck failed.”

Additionally, when the robot failed to detect chuck because it could not lift torso (first node at depth 7), the robot will answer “I am retrying for attempt 1 to go insert large gear. I could not detect chuck because lift torso failed.”

If asked in the screw picking and place subtasks, the answers would be very similar.

4.6.5 Taxi Domain: A Navigation Task

While this work was originally designed for the context of the kitting task, we show here the potential of this work in a non-manipulation domain. In this modified taxi domain [53], a taxi agent is tasked with picking up and dropping off a passenger while accruing into the fewest tolls possible.

We formalize this task as a deterministic Markov Decision Process, $MDP := (S, A, T, R, \gamma, S_0)$, where S and A are state and action sets respectively, $T : S \times A \rightarrow S$ is the transition function, $R : S \times A \times S \rightarrow \mathbb{R}$ is the the reward function, $\gamma \in [0, 1]$ is the discount factor, and S_0 is the initial state. Tolls provide negative reward, and delivering a passenger at the goal provides a large positive reward. The agent has an optimal policy $\pi^* : S \rightarrow A$ that maps states to actions that optimize the reward given an infinite horizon. An instance is shown in Figure 30.

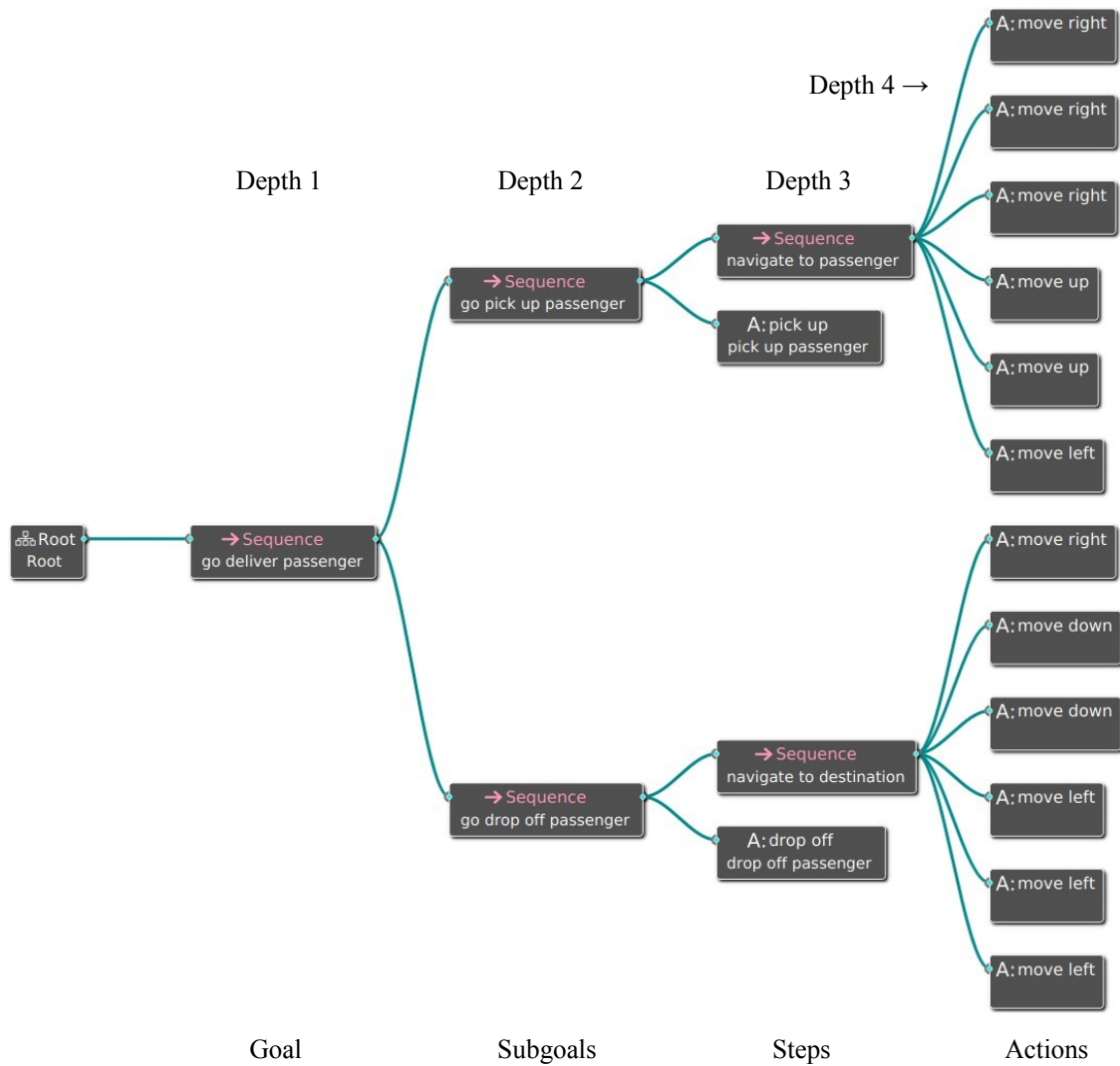


Figure 31: The behavior tree representation of an optimal policy in the non-manipulation taxi domain. Given that it is less complex than a kitting subtask, there is no need for simplification.

The Taxi Domain is implemented in PyGame²⁰ and the `simple_rl` framework [3]. We trained the agent using value iteration [151].

The optimal policy π^* for the instance is represented in the behavior tree in Figure 31. With domain-specific goal and subgoals, the tree is semi-automatically generated by implementing listeners to state-action switches. For example, switching from moving in one of four directions to picking up the passenger will result in generating the `navigate to passenger` sequence.

The goal here is to deliver the passenger, which consists of two subgoals: go pick up passenger and go drop off passenger. To pick up the passenger, the agent first navigates to the passenger by moving all the way right (3 `move right` actions) and all the way up (3 `move up` actions), and takes a `move left` action before finally picking up the passenger. To drop off the passenger, the optimal agent takes the reverse path back to the destination. By taking the rightmost path, the agent avoids the toll in row 2, column 3.

As noted in the caption of Figure 31, the behavior tree representation is already in its simplified form, with no need for framing, such as removing decorator nodes. However, as shown below, the algorithms still apply without the use of decorator nodes as well as input and output ports. This shows that the behavior tree does not have to be complex in order to leverage the algorithms for explanation generation.

Similar to the large gear insertion task, we first examine the first action node `move right` at depth 4 in Figure 31. The algorithms can answer Q1 to Q5:

Q1. What are you doing?

I move right.

Q2. Why are you doing this?

I move right in order to navigate to passenger.

²⁰<https://www.pygame.org/>

Q3. What is your subgoal?

My subgoal is to go pick up passenger.

Q4. What is your goal?

My goal is to go deliver passenger.

Q5 (subgoal). How do you achieve your subgoal?

To achieve the subgoal “go pick up passenger”, I need to do 2 steps. 1. navigate to passenger. 2. pick up passenger.

Q5 (goal). How do you achieve your goal?

To achieve the goal “go deliver passenger”, I need to do 2 steps. 1. go pick up passenger. 2. go drop off passenger.

A different and more interesting example is the execution of the `pick up` or `drop off` action node at depth 3, which is not the deepest. Here we will take the example of the `drop off` action node – the case for the `pick up` node is very similar. The following are the answers from the algorithms:

Q1. What are you doing?

I drop off passenger.

Q2. Why are you doing this?

I drop off passenger in order to go drop off passenger.

Q3. What is your subgoal?

My subgoal is to go drop off passenger.

Q4. What is your goal?

My goal is to go deliver passenger.

Q5 (subgoal). How do you achieve your subgoal?

To achieve the subgoal “go drop off passenger”, I need to do 2 steps. 1. navigate to destination. 2. drop off passenger.

Q5 (goal). How do you achieve your goal?

To achieve the goal “go deliver passenger”, I need to do 2 steps. 1. go pick up passenger. 2. go drop off passenger.

As the behavior tree does not have any input or output port dependencies, the dynamic behavior insertion as subgoal for the taxi domain is trivial. We can insert the 2 subgoals with the following questions:

Can you go drop off passenger?

Can you go pick up passenger?

Using Algorithm 8 and 9, the corresponding subgoals represented by subtrees are inserted after the current subgoal.

4.6.6 Explaining Divergences Between Behaviors

When a robot is executing its behavior tree, people’s interpretations might be different, since humans tend to impute their own beliefs onto others [125]. After just a few executions, people will have often already formed a mental model of the robot’s behavior. When this mental model differs from the robot’s internal model, it is important to clarify where the divergences are and explain this discrepancy to avoid confusion.

An example can be drawn from the taxi domain. Recall that Fig. 31 shows an optimal policy where the toll in row 2, column 3 in Fig. 30 was avoided in order to maximize the reward. After a few passenger deliveries in different environments, a person may still think that taking a *single* extra toll at row 2, column 3 is more reasonable than detouring through column 4 taking *three* extra

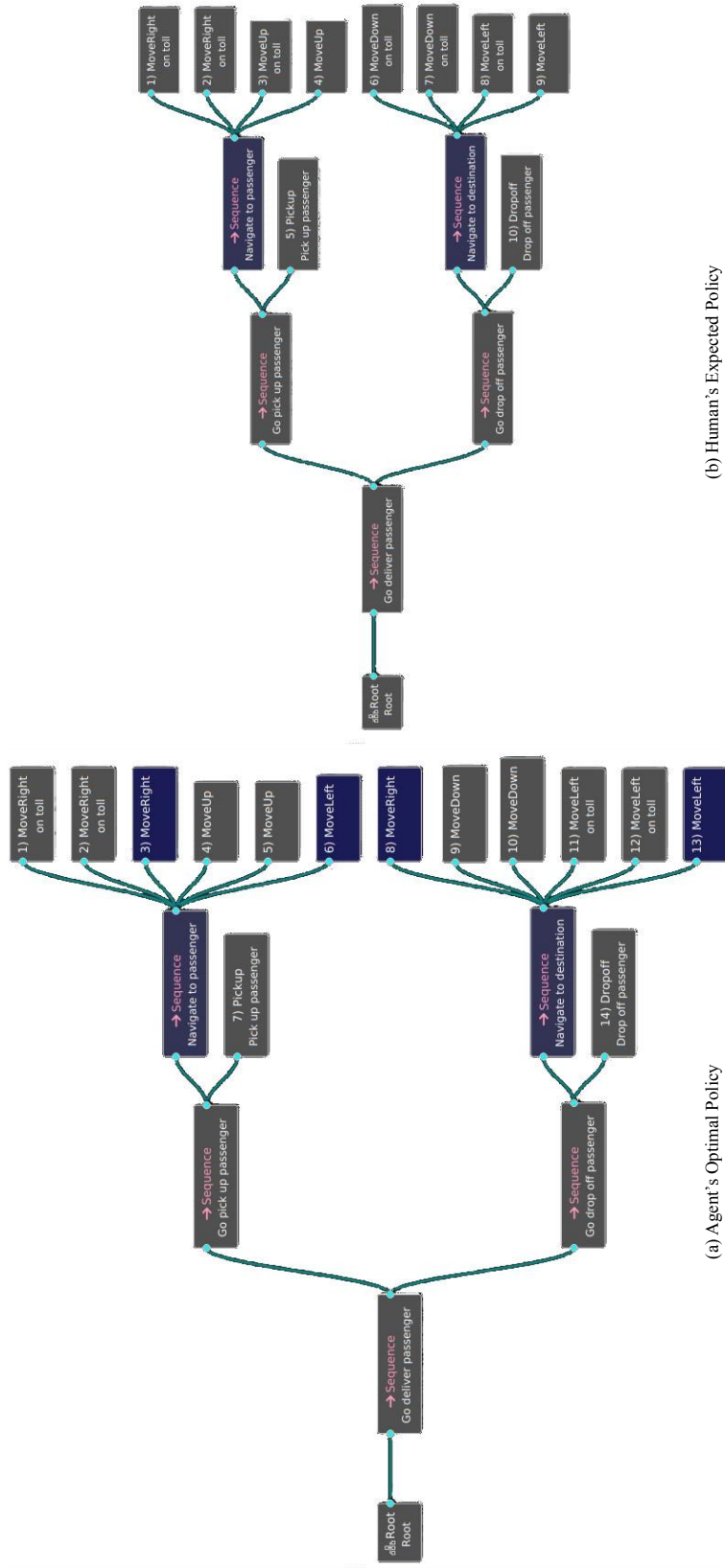


Figure 32: An agent's optimal policy vs. a human's expected policy for the environment in Fig. 30. Divergences in actions (some leaf nodes) between the two policies are colored in dark blue and their immediate parents (some sequence nodes) are colored in blue-gray.

steps to avoid the toll. They may not understand that 1 toll is more costly than 3 steps. In cases like this, it is beneficial to show where the divergences are to improve understanding.

Behavior trees can reveal such divergences automatically by presenting the robot’s actual behavior and the human’s expected behavior for easy comparison. As shown in Fig. 32, the behavior tree representations of the two are placed side by side and the differences are highlighted in blue. The divergences were found using the sequence comparison algorithm proposed by Wu et al. [171]. The action sequences of the same subgoal between the two policies are compared by calculating the edit distance and Longest Common Subsequence (LCS). Note that this visual comparison can also be used to differentiate behavior between multiple robots to improve comparative understanding, and we are currently implementing it to complement to the visualization in Fig. 32.

4.7 Limitations and Future Work

To the best of our knowledge, we are the first to explore Behavior Trees for robot explanation. We understand that there are still several limitations to the proposed work. We believe BTs are promising, but still require more work to satisfy the needs of a robust robot explanation system.

BTs have the ability to allow custom decorator nodes but framing cannot natively handle them, which might need to be explained when answering users’ questions. A workaround solution is to have special cases in the algorithms, as in line 3 in Algorithm 2 and line 3 in Algorithm 5.

Also, to date, we have focused on the technical aspects, evaluating the systems with case studies for subtasks in the kitting task, a gear insertion task for machining, and within the taxi domain; a user study is still needed with non-expert robot users to evaluate the level of understandability and the perception of the causal information in the answers to Q2–Q5. Having said this, our focus has been contributing robot generation algorithms in the scope of behavior trees, which paves the road to a robust explanation generation accounting for more human factors.

While we have demonstrated the use of our algorithm in manipulation domains and the taxi domain, it remains relatively unknown how to support other non-manipulation tasks. For example,

in a multi-robot system or a fleet of autonomous cars, robot explanations are increasingly needed given the increased complexity. However, given the complexity of robot manipulation and a sample use in a non-manipulation domain, we believe that the proposed algorithms can cover additional use cases, which will be explored in future work.

Another future goal is to investigate the questions being answered in this work; they are currently only analytically grounded but need to be validated in a user study. Currently, there is also a lack of research on what questions people would ask to get causal information on the behaviors that a robot is exhibiting. Like humans, are the “why” questions the only things humans seek for causal information from a robot’s behavior? What about the “how” questions, such as “how do you achieve the goal”? What specific questions do people ask a robot generally? What causal questions do people ask a robot? These remain open research questions.

A drawback of BTs is that they do not generate a smooth, continuous arm trajectory if each waypoint is encapsulated in a behavior node separately. The recent development of the MoveIt Task Constructor (MTC) [73] looks promising on this front. This library is a wrapper around the commonly utilized features of MoveIt such as motion planning and obstacle avoidance, but facilitates being able to track the current progress of execution, as well as makes specifying complicated tasks, such as pouring, easier. In MTC, manipulation primitives are represented as stages. Each stage can be linked together with other stages to form complete manipulation behavior. Examples of stages can be commands such as motion planning and moving the end effector to a specific pose or direct commands to the gripper joints. The success of each group of stages can propagate upwards as a reason for the success or failure of the behavior. For example, failure to reach a designated goal because of a detected collision would result in the behavior failing, because there is an obstacle blocking the path. These fine details, if used in the replies to the explanation questions, can further improve the understanding of the robot and its behaviors.

As MTC opens up the possibility for explanations at the low-level motion planning, we also need to explain low-level path planning in navigation, e.g., the popular ROS navigation stack [109].

Similar to MTC, we can enable the robot to explain if there is an obstacle on the ground blocking its path. Fortunately, we recently learned that the navigation stack in ROS 2 uses BTs to represent navigation behaviors [107], which paves the way for our explanation algorithms to be integrated into the navigation stack.

Last but not least, because BTs are also sequential, BTs do not support some properties of behavior execution that humans and animals are born with, such as pausing and reversing actions, which are the capabilities that a robot needs to have to achieve more natural, human-like robot explanation in certain circumstances.

4.8 Conclusion

In this paper, we have demonstrated the use of Behavior Trees (BTs) for high-level robot explanations. We proposed framing BTs into a set of semantic sets, i.e., {the goal, subgoals, steps, actions}, and applied them to screw picking, screw placing, and large gear insertion manipulation tasks as well as the taxi domain. We contributed algorithms to generate robot explanations, specifically giving answers to questions seeking causal information, which is what humans commonly seek from human explanations. We also described an algorithm that finds self-contained behavior based on the matching node from what is being asked, then inserts it after the current subgoal that the robot is achieving. We hope our work inspires other researchers working towards the goal of transparent and trustworthy robots, and paves the road to a robust robot explanation system.

In the chapter after the next chapter, we will use the screw picking and placing tasks as well their behavior tree representation to investigate how robots can aid inference-making.



Figure 33: The projection of perception results: the detected objects (white and green) and the object to be manipulated (green). Using our implementation of projection mapping, researchers and practitioners can enable a robot to accurately externalize internal states for explanation. A video is available at <https://youtu.be/S0z9e2gUrEA>.

5 Projection Mapping Implementation²¹

5.1 Introduction

Traditionally, because of the distinct embodiment feature of physical robots, human-robot interaction (HRI) researchers have been focused on how to enable robots to communicate their intention through non-verbal means that are typical among humans, most notably pointing through eye gaze [118, 6], and arm movement [56, 100]. Light as an indication has also been used [152].

These non-verbal cues found in human life play an important role in supporting and improving communication [5], but can also cause confusion. For example, how can a robot communicate which objects it has detected and which one is it going to grasp? In cases like these, using eye-gaze and/or pointing with its arm or end effector can be vague, especially in a clustered environment (e.g., for four clustered objects on a table). Even with verbal explanations, these gestures could be underspecified, requiring follow-up questions for clarification.

²¹This chapter appears in a paper [83] jointly authored with Alexander Wilkinson, Jenna Parrillo, Jordan Allspaw and Dr. Holly A. Yanco. Please see Publication 5. Thanks to Analog Devices (ADI) and MassRobotics for organizing the ADI Sensor Fusion Challenge and providing funds to purchase the sensors.

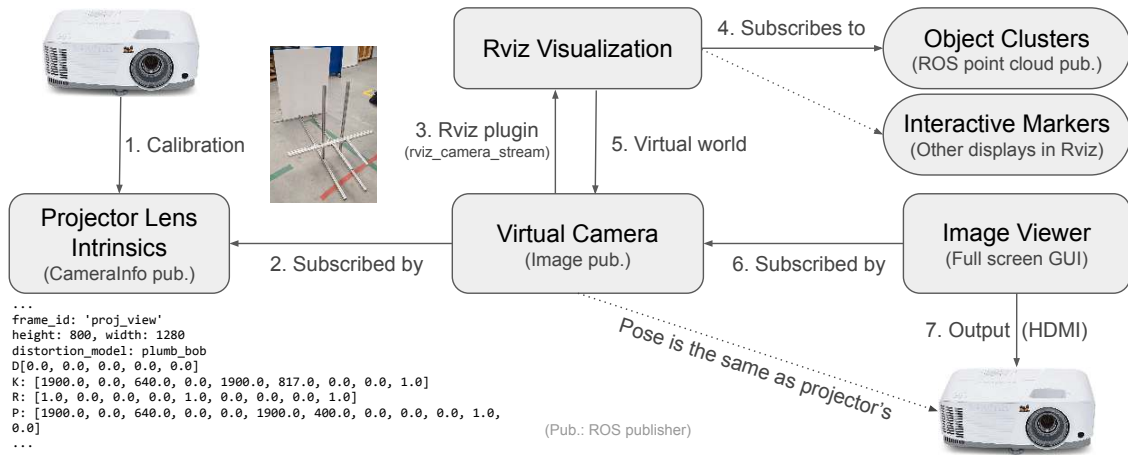


Figure 34: High-level diagram for our projection mapping implementation. With the projector lens calibrated, a virtual camera – placed in Rviz with the same pose as the projector in real world – subscribes to the camera intrinsics so it can output an image of objects visualized in the virtual world in Rviz to the projector to reflect the perceived objects. See Section 5.2 for more details.

In this chapter, we present a tool for implementing projection mapping using an off-the-shelf projector, including the high level architecture and low level technical details, in order to project perception results directly onto non-flat objects of interest in the environment. We also describe a concrete robot and hardware platform as an example of the tool’s use, even though the technique is robot-agnostic. Note that while we focus on projection mapping with the same object as input and output, projection mapping can be applied more broadly to arbitrary objects and targets [75].

The work in this chapter laid the foundation for the human-subjects study to be discussed in the next chapter.

While projection mapping in robotics is not a new idea, the implementation effort has been missing and thus not as accessible as implementing arm movement or eye gaze through head movement. This gap effectively blocks HRI researchers from conducting human-subject studies to investigate the effects of accurate externalization through projection mapping or by comparing it to other methods. In addition, this work can also help robotics and AI researchers to externalize the output of their computer vision algorithms for a better understanding of their algorithms.

5.2 Overview

Figure 34 shows the high-level architecture of our implementation. Essentially, we set up a virtual camera with the same lens intrinsics in the same pose as the projector's in the physical world in the ROS visualization software, Rviz. We then publish perception results in point cloud clusters and the object point cloud to be manipulated, then add point cloud visualization in Rviz. Finally, we have an image viewer in a full screen GUI to output the image that the virtual camera sees to a projector. We implemented everything in ROS [138]. All of the files are available on GitHub²², including a sample Rviz config file, sample point clouds in pcd format, and the launch files for publishing lens intrinsics and the pose of the projector.

Our projection mapping system functions on the principal that a projector is the dual of a camera. If a camera is thought of as a map from 3D world coordinates to 2D image plane coordinates, then a projector is a map from 2D image plane coordinates to 3D world coordinates. In a camera, each light ray from the world passes through the lens and hits the sensor. In a projector, each light ray passes through the lens and hits a surface in the world.

By using a virtual camera with the same intrinsics and extrinsics as the projector, we can then map points from the virtual world's 3D space to 3D space in the real world.

5.3 Projector Selection Consideration

While any projector should theoretically work as long as we calibrate it to get the lens intrinsics as detailed below, there were a couple of factors that made us choose the ViewSonic PA503W projector in our implementation. In the common use case where projectors are used to watch movies and show presentation slides, the room lighting is often switched off or dimmed to make the projection more bright and thus more legible. However, robots often operate indoors with lights on and not dimmed, so the consideration here is to make the projection visible and legible even under bright light conditions.

²²https://github.com/uml-robotics/projection_mapping



Figure 35: We calculated the intrinsics of our particular projector by mounting it perpendicular to a posterboard at a fixed distance. See Section 5.4 for more details.

There are three contributing factors: brightness, contrast and the projection technology. The standard measure of brightness is the ANSI lumens value, which is a measure of the total light output per unit of time. For reference, our PA503W produces 3,800 ANSI lumens of light. Contrast is expressed by a ratio between the darkest and lightest areas of an image, with our PA503W being capable of a 22,000:1 contrast ratio. Finally, there are two popular projection display technologies: Liquid Crystal Display (LCD) and DLP [92]. As a brief summary from the work by Hornbeck, DLP is based on a Digital Micromirror Device (DMD), and it has a higher brightness than LCD because DMD has a reflective and high-fill factor digital light switch, as opposed to active-matrix LCDs, which are transmissive and thus the generated heat cannot be dispatched well. See [92] for more detail.

5.4 Projector Calibration

We modelled the projector with a pinhole lens model after determining the focal length f and principal point (c_x, c_y) . Both of these values were calculated manually by mounting the projector perpendicular to a flat surface at some fixed distance (e.g., 35), marking the corners of the image,

and then using the projective equation:

$$f = w \frac{Z}{W} \quad (1)$$

where w is the width of the projected image in pixels, W is the width of the projected image in meters, and Z is the distance from the projector to the image plane in meters.

The principal point (c_x, c_y) can be calculated given X and Y , the respective distances from the origin to the intersection of the optical plane and the optical axis in meters:

$$(c_x, c_y) = \left(w \frac{X}{W}, h \frac{Y}{H} \right) \quad (2)$$

where h is the height of the projected image in pixels.

These values for f , c_x , and c_y were then placed into an intrinsic camera matrix K :

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

With this matrix calculated for our projector, we then publish it in a ROS CameraInfo message²³ and set up a virtual camera in RViz (see Section 5.5) that subscribes to the CameraInfo message and is located at a TF transform corresponding to the pose of the physical projector. The CameraInfo message for the ViewSonic projector and the details of this process are in our code²⁴.

It is worth noting that measurement error may accumulate during this manual process. Since consumer projectors are typically designed with little radial or tangential distortion, we did not explicitly model lens distortion. In practice, the projection does not deviate much from the real objects, but it could be improved by more accurate projector-camera calibration methods [119].

²³https://docs.ros.org/melodic/api/sensor_msgs/html/msg/CameraInfo.html

²⁴https://github.com/uml-robotics/projection_mapping/blob/master/projector_camera_info.yaml

5.5 Virtual Camera

We developed `rviz_camera_stream`²⁵, an Rviz plugin that outputs an image of what a camera sees in the Rviz virtual world. We refer this as the *virtual camera*.

Represented in a ROS TF frame [63], the pose of the virtual camera – placed in the virtual world in Rviz – is the same as the pose of the projector in the real-world. Together with the fact that the virtual camera pose is in the same transform hierarchy as a perception sensor, we are able to transform a point cloud or any other visualizations from the perception sensor’s frame to the virtual camera’s frame. This method allows the projection to be in a projector’s view point of what the perception sensor sees.

To have the same optical properties as the projector, the virtual camera subscribes to a ROS topic with the `CameraInfo` message mentioned in the previous subsection. This ensures that, when objects are projected back to the real-world, the projected objects match the object’s physical shape and size.

Finally, because this virtual camera resides inside Rviz, it is able to see everything being visualized in Rviz, such as interactive markers [74], navigation maps, or point clouds. For example, point clouds are seen by the virtual camera and projected in Figure 33.

5.6 Projection Output

Because the `rviz_camera_stream` package publishes the image of what the virtual camera sees, we use the image viewer from the `image_view` ROS package to subscribe to the image topic and output to the projector through HDMI. Note that the image viewer is in full screen by enabling and using the “Toggle full screen” keyboard shortcut in the keyboard setting of Ubuntu’s Settings software.

²⁵https://github.com/ucl-robotics/rviz_camera_stream; thanks to Lucas Walter (<https://github.com/lucasw>) for his contribution.



Figure 36: A sample use, where a projector is mounted onto a Fetch robot via a custom hardware structure attached to its upper back and a turret unit to pan and tilt the projector. However, the projection mapping technique is robot agnostic and the projector does not have to be attached to the robot. See Section 5.7 for more details.

5.7 Hardware Platform

As seen in Figure 36, we have demonstrated an implementation on a Fetch robot with a custom structure to mount a projector. While it would have been easier to attach the projector to the robot’s head, the neck may not have had enough torque to bear it, so we chose to attach the structure to the robot’s upper back. In order to pan and tilt the projector, we attached a ScorpionX MX-64 Robot Turret Kit²⁶.

Despite using a Fetch robot, the projection mapping technique we describe is robot agnostic. Previously, we also applied it on an assistive robot [160]. The only requirement is that there is a

²⁶<https://www.trossenrobotics.com/p/ScorpionX-RX-64-robot-turret.aspx>

transform frame for the projector so it is integrated into the transform hierarchy of a robot. In our case, we created two frames: one for the projector lens and another for the attachment point at the bottom of the projector. However, this requirement does not necessarily mean that the projector must be attached to the robot, but instead that it must be co-located with the robot in the operating environment. For example, if one would like to use projection mapping with an industrial robot arm, the projector can be placed nearby on a structure as long as its lens is pointing at what the perception sensor is targeting. By doing so, a robot arm will also not block the projection during manipulation, as careful readers may notice in the accompanying video. Otherwise, one can add a cylinder collision object to model the line of sight to mitigate blockage.

5.8 Conclusion

In this chapter, we have detailed our approach to and implementation of projection mapping in robotics. Four major components were discussed: projector consideration and calibration, a virtual camera in Rviz, projection output, and a sample hardware platform.

To implement projection mapping, a roboticist can purchase an off-shelf projector, calculate a coordinate frame after installing it, calibrate its lens to get the lens intrinsics, and publish it in an encapsulated `CameraInfo` message. Then the roboticist would set up a virtual camera using the `rviz_camera_stream` Rviz plugin and subscribe to the `CameraInfo` message. Then the roboticist could add point cloud visualizations or any other displays (e.g., interactive markers) in Rviz and have the projector point them either manually or using a turret unit, and finally use an `image_viewer` to output the image from the virtual camera to the projector.

In the next chapter, we will use this projection mapping technique for manipulation and navigation projections.

6 Communicating Missing Causal Information of Past Actions²⁷

6.1 Introduction

Current research has focused on investigating in-situ explanations of robots' current actions, which happen at or around the moment. For example, researchers recently have investigated the use of images of kitchen cleaning tasks with a description explaining the robot's current behavior [157]. Other research includes a robot explaining why it undesirably blocked a TV while a person is watching TV at the moment [149]. Technical approaches include explanation generation of robot's current behaviors using function annotation in assembly lines [87], an encoder-decoder model [43], and behavior trees [82].

Yet it remains relatively unknown how a robot would communicate explanations of its past actions. Providing explanations for past behavior is especially interesting because the environment might change after the robot's actions. For example, objects might be moved after manipulation tasks or obstacles on the floor might be removed by people after navigation tasks (e.g., a wet floor sign removed by cleaning staff once the floor becomes dry). These environment changes lead to some missing causal information which may confuse people when the robot later explains its behavior with references to objects that were only present in the past.

Robots must account for these scenarios and provide indications to help people infer the missing causal information that might not be present while explaining. Indeed, researchers in psychology have found that humans hope to gain causal knowledge from explanations by others [105, 22]. When the causal knowledge is present, understanding is improved and "people can simulate counterfactual as well as future events under a variety of possible circumstances" [108].

To expand our knowledge on how a robot can help people to infer past missing causal information, we conducted an experiment online through Prolific [132]. Participants watched videos of a Fetch robot [170] replaying its past actions in a collaborative mobile kitting task, consisting of

²⁷The work in this chapter was submitted to ACM Transactions on Human-Robot Interaction in 2021. Vittoria Santoro and Jenna Parrillo contributed to the replay implementation.



Figure 37: A mobile manipulation task environment in which we investigated how could the robot provide indicators to past missing causal information. The robot is supposed to pick different gearbox parts, including the gearbox bottoms on the table it is facing, take them to the caddy table on the left, and deliver the caddy to the bottom right table for an assembly worker to assemble a gearbox.

both manipulation and navigation.

In the task (Figure 37), the Fetch robot is supposed to help a worker to assemble gearboxes: the robot picks one of the gearbox bottoms on a table, navigates to a caddy table, and places the gearbox bottom into a caddy. The caddy has three compartments: one big rectangular section and two small square sections (See Figure 37 left). The robot has to put the gearbox bottom into the bigger compartment because the object does not fit into any of the other two small compartments. The task was originally designed for the FetchIt Mobile Manipulation Challenge and there are different parts for the robot to pick. For more details, please see our previous paper [79].

In the experiment, we considered three highly motivated scenarios where a robot needs to replay its past behavior to help people to infer missing information because the robot’s explanations are unexpected – inconsistent with what participants observe. The scenarios below are narrated from the worker’s perspective and presented to participants throughout the experiment:

1. Picking Failure Scenario. “As shown below (Figure 37), a robot is helping you to assemble

gearboxes. It can drive itself, pick, and place a gearbox bottom into a caddy. One day, you leave your work area for a few minutes. When you return, you notice there are still 2 gearbox bottoms on the table. You ask the robot if it just picked up a gearbox bottom, it says yes. You are confused, and ask the robot to replay its past behavior. Then you see the robot was grasping a large wood chip torn up from the tabletop.”

“In the video, the robot replays its past behavior: picking up a large wood chip that the robot thought is a gearbox bottom. At replay time, the wood chip is gone already. Can you figure out where the large wood chip was before?”

2. Navigation Scenario. “One day, another worker nearby told you the robot didn’t go straight to the caddy table. But you see the robot every day, and it does go straight to the caddy table every time. You are confused, and ask the robot to replay its past behavior. Then you see the robot was actually avoiding an obstacle on the ground.”

“In the video, the robot replays its past behavior: driving itself to the caddy table. At replay time, the obstacle on the ground is already gone. Can you figure out where the obstacle on the ground was before?”

3. Placing Scenario. “One day, you hear a gearbox bottom dropped onto the floor behind the caddy table. You ask the robot if it just put a gearbox bottom into a caddy, and it says yes. You are confused, and ask the robot to replay its past behavior.”

“In the video, the robot replays its past behavior: putting a gearbox bottom into a section into the caddy. At replay time, the gearbox bottom is already gone. Can you figure out which section of the caddy that the robot tried to put the gearbox bottom into?”

As seen from the questions, there are three missing pieces of causal information due to environmental change which the robot should indicate at replay time.



Figure 38: The failure scenario that inspired the first picking scenario: A Fetch robot misrecognized a torn up wood chip near the top-right corner of the table as a screw.

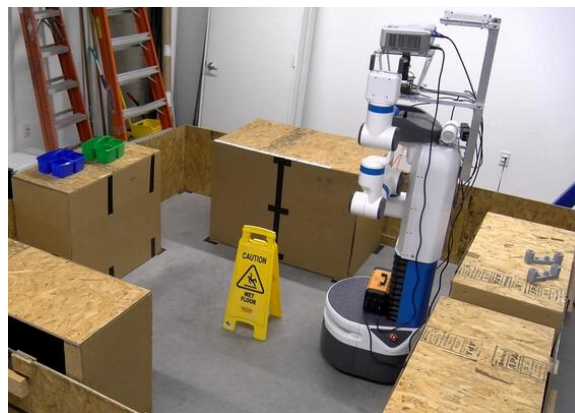


Figure 39: The original scene where the robot avoided a ground obstacle, the yellow wet floor sign, to navigate to the caddy table to the left. At replay time, the yellow wet floor sign is gone. Key videos frames for the replay video without the sign is shown in Figure 42.

1. For picking, because a large wood chip was recognized as gearbox bottom and the chip is gone, the robot needs to indicate where it has thought the gearbox bottom was.

This is inspired by a real-world failure where the robot once misrecognized a wood chip as a smaller screw, shown in Figure 38. Because the screws were placed inside a container hiding the screws, we used the larger gearbox bottoms and placed them directly on the table.

2. For navigation, the robot has to take a seemingly non-optimal navigation path because there was a ground obstacle – a wet floor caution sign – between the two tables during navigation (See Figure 39).

3. For placing, the robot should indicate which caddy compartment the gearbox bottom was placed into, so a human can understand that a gearbox bottom does not fit a small caddy section so the gearbox bottom slipped and dropped onto the floor.

To help people to discover the missing causal information, we manipulated seven methods that the robot used to indicate relevant actions in the mobile kitting task. Our base condition was a physical movement replay (head, arm and base movement) where the robot would physically replay its past actions. We tested this base condition against conditions for three communication methods including speech, projection, as well as with both speech and projection. These three conditions were also tested in combination with physical replay.

In a questionnaire, we asked participants whether and when they have inferred the missing information – the locations of the misrecognized object, the ground obstacle, and the section of the caddy the object was placed into. Participants were also asked about their confidence in their inference answers, participants' mental workload, and robot trust.

6.2 Hypotheses

Driven by the observational learning work in Section 2.6 and particularly that demonstrator's intention aids causal inference, we formalized the following hypotheses.

Hypothesis 6.1 – Effective causal inference with verbal markers; Adding verbal markers to relevant actions in a robot's action sequence will help people to effectively infer the causality of the robot's behavior, measured by subjective measures about the three missing causal information: where the gearbox bottom was, why the robot did not choose the straight path, and which caddy compartment the gearbox bottom was placed into. This hypothesis also tries to validate the human intentionality study in [68] that intentional actions with verbal markers, such as "here" and "there", help to understand causality.

Hypothesis 6.2 – The same effectiveness of causal inference with projection markers vs. verbal markers; Adding projection markers to relevant actions will have at least the same

effectiveness to infer the missing causal information as verbal markers. The measure for this hypothesis is the same as H6.1.

Hypothesis 6.3 – Faster and more accurate causal inference with projection markers; Adding projection markers will make the causal inference faster than verbal markers, measured by when participants find the three pieces of missing information. As projection directly provides the causal information to the robot’s operating environment, we expect projection is at least the same as verbal markers in terms of causality inference.

Hypothesis 6.4 – The same workload in both verbal and projection conditions; Having projection markers will have the same amount of workload for the inference tasks, measured by subjective measures.

Hypothesis 6.5 – A robot is more trustworthy with projection markers; Because projection markers have the potential to infer missing causal information faster, we believe people will trust the robot more, measured by subjective measures.

Hypothesis 6.6 – There will be less workload when presented both verbal and projection markers; Inspired by the multiple resource theory by Wickens [168] and the visual buffer effect [150], we believe that when more channels are used to convey the causal information, it makes the inference easier, and thus requires less workload. The measures will be subjective and the same as H6.4.

6.3 Experiment Design

The experiment followed a between-subjects design. In each condition, different participants watched three videos of a mobile manipulation task and completed a survey.

6.3.1 Task

In the videos, the robot replayed three subtasks in a mobile kitting task: it first tried to pick a gearbox bottom, navigated to the caddy station, and placed the gearbox bottom into a caddy.

As mentioned earlier, the robot navigated in a detoured rather than a straight path during the replay as there was a wet floor caution sign along the way. Additionally, the robot replayed its gearbox bottom grasping without any gearbox bottom in its hand because it treated a large wood chip torn from the table as a gearbox bottom. The large wood chip was already gone at replay time. Except for these two missing causal information, because no gearbox bottom was physically placed into a caddy compartment, the compartment position was also missing.

At replay time, we used verbal markers and projection markers to indicate the missing causal information: which gearbox bottom the robot was grasping, why the robot detoured during navigation, and which caddy compartment the gearbox bottom was placed into.

The mobile kitting task was performed in an enclosed arena. For participants to see the whole arena and the ground obstacle projection in the videos during the picking and navigation replays, we set up a camcorder on a tall tripod sitting on a table at the near right corner outside of the arena. Specifically, the distance from the lens of the camcorder to the floor was around 1.9 meters (7 feet 8.5 inches). The placement was intentional to cover the wide field of view of a human's eyes. In the placing replay, we placed the camcorder to the left of the robot outside of the arena to get closer to the caddy table.

6.3.2 Study Conditions

There are seven conditions in this experiment, designed to show different approaches to indicate missing causal information during the replay of the robot's past actions. As we describe each condition below, Figure 40 to 44 show the key video frames from the videos of the conditions.

1. Replay. During this condition, the robot replayed all actions in the action sequence of the task without any verbal or projection indication: all head, arm, and wheel movements. No explicit indications of causality were expressed by the robot. We have this because this is very similar to introspection which requires humans to investigate thoroughly.

2. Replay-Say. In addition to replaying all actions, the robot speaks regarding the miss-

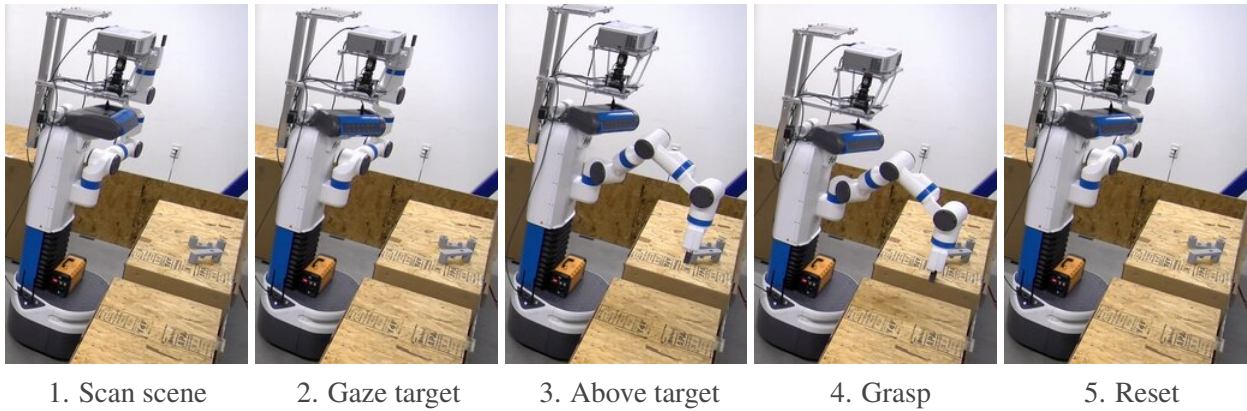


Figure 40: Snapshots from the picking videos with physical replay. During the picking task, the robot mistreated a wood chip as a gearbox bottom on the right edge of the table. Participants were asked to infer where the robot has picked. Verbal indicators are in Table 7. Projection photos are in Figure 41. There was no arm movement in the Say, Project, and Project-Say conditions.

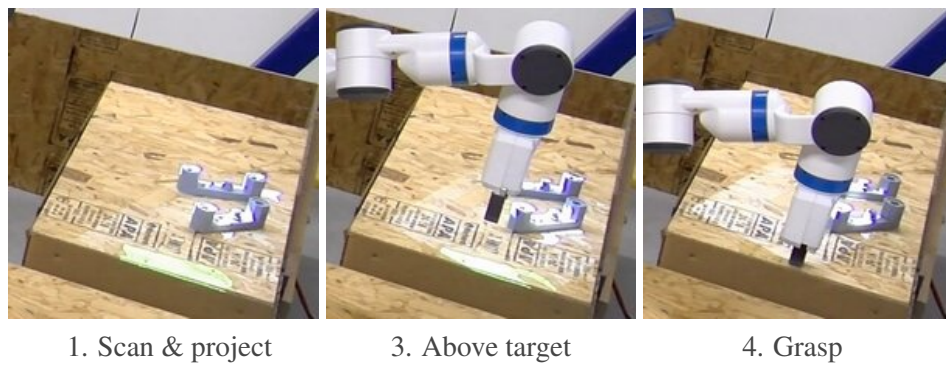


Figure 41: Tabletop projections in the picking videos. Recognized objects are projected in white; The target object to be picked is in green. These three snapshots are corresponding to the first, third, and fourth snapshot in the previous figure (Figure 40).

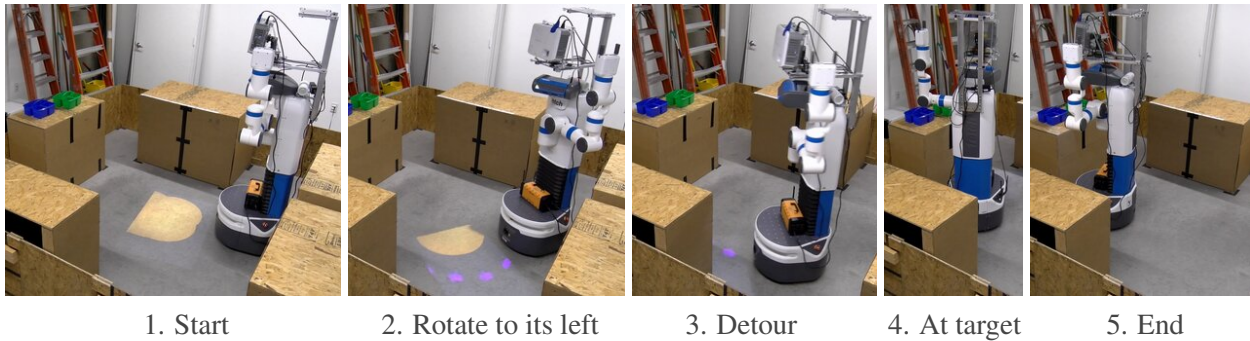


Figure 42: Snapshots from the navigation video in the Replay-Project condition, where participants need to infer where the ground obstacle (a wet floor sign) was. The yellow projection consists of multiple 3D spheres of point clouds from the robot’s base laser scan. Purple arrows indicate the robot’s detour path. Non-projection conditions had no projection on the ground. For conditions with speech, its text is shown in Table 7. In the Say, Project, and Project-Say conditions, the robot’s base did not move.

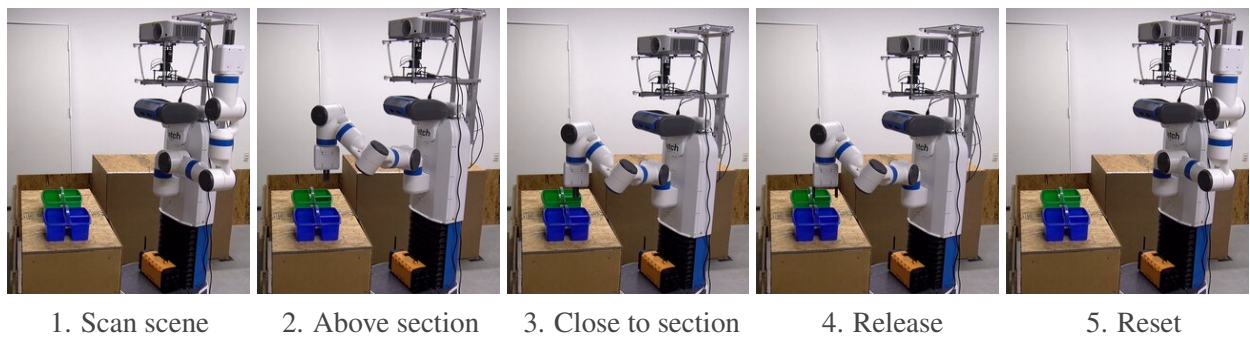


Figure 43: Snapshots from the placing video in the Replay condition, where participants need to infer which section of the caddy that the robot placed into. Again, for speech condition, the text is shown in Table 7. Projection photos are shown in the next figure (Figure 44).

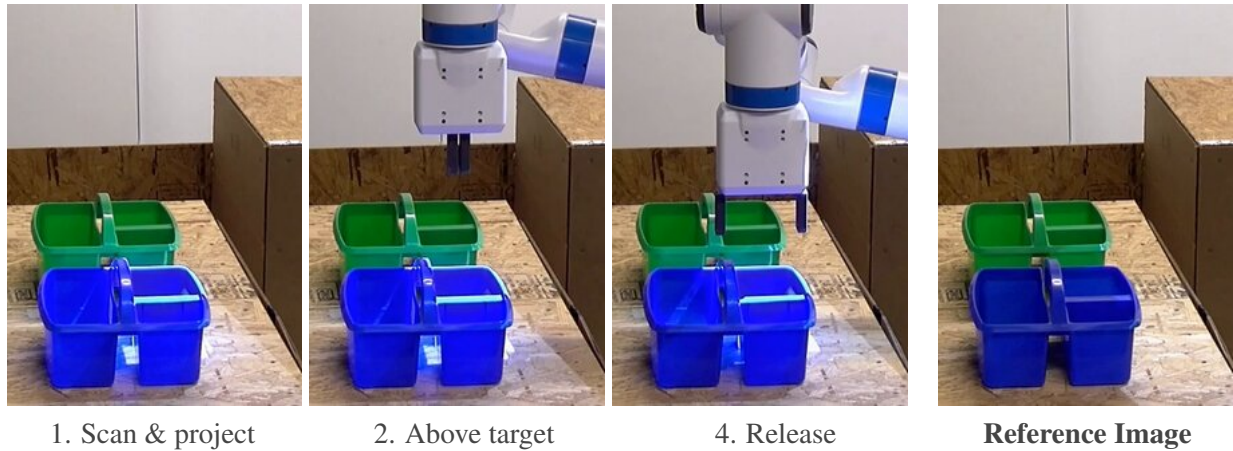


Figure 44: Tabletop projections in the placing videos. The last snapshot is from the Replay condition video and is indented to easily differentiate the projection from the photos to its left. The first three snapshots are corresponding to the first, the second, and the fourth snapshot in the previous figure (Figure 43) The rightmost reference image has been provided here to show what the image looks like without any projection, to allow for differentiating where the projection is on the left three figures.

ing causal information during relevant actions to indicate the location of the gearbox bottom, the ground obstacle, and the caddy compartment, as shown in Table 7. The earliest relevant actions are chosen to avoid ambiguity.

As seen in Table 7, we have chosen simple words in the speech for participants to easily understand. The speech was generated using Google Cloud Text-to-Speech²⁸: WaveNet [130],

²⁸<https://cloud.google.com/text-to-speech>

Table 7: Causal Verbal Markers And Their Timing

| Speech | Timing |
|---|---|
| “Ok. I picked up a gearbox bottom from here.” | Robot’s gripper is over the target object (Third photo in Figure 40). |
| “Ok. I didn’t go straight to the caddy table because there was something on the floor in front of me on my left.” | Before robot starts driving itself. |
| “Ok. I placed the gearbox bottom into the near right section of the caddy.” | Robot’s gripper is over the compartment. |

specifically en-US-Wavenet-D. We lowered the speech speed to 85%, as suggested by [111], to counter the noise from the air conditioning system at the ceiling in our facility. Because the volume from the robot, Fetch's base speakers was very low, we bought a 20W JBL FLIP 5 Bluetooth speaker²⁹ and placed it behind the neck of the robot. We chose to buy a white one to match the Fetch robot's primary color and, thus, it makes the speaker less noticeable.

3. Replay-Project. The robot replayed all actions and, instead of speaking, projected the causal information during relevant actions, the perception result and manipulation target back to the operating environment during picking, arrows of its navigation path as well as the 2D projection of multiple spheres representing the laser scans of the ground obstacle, and a cubic projection to represent the space that a caddy compartment occupies.

This condition is interesting because projection is not a human capability and we want to explore the use of projection mapping, a more salient effect, inspired by the action effects discussed in Section 2.6. As seen in hypothesis 6.2, we expect it to have the same effect as the verbal markers.

4. Replay-Project-Say. This condition includes both verbal and projection indicators. The combination is inspired by our previous study in Chapter 3, which shows verbal explanations is needed in addition to the non-verbal cues. It is also inspired by Multiple Resource Theory [168], which states that a cross-modal interface, using modalities that reside in two different channels, has advantages over an intra-modal interface, using modalities that reside in the same channel.

The first four conditions above have physical movement replay. The remaining three below do not have replay, to avoid replay being a confounding factor affecting participants' responses.

5. Say. This approach is to only use verbal markers to indicate the gearbox bottom, the ground obstacle, and the caddy compartment.

6. Project. This condition is only using projection mapping to project the perception result back to the operating environment for the indication.

7. Project-Say. This condition combines both verbal and projection indicators but without

²⁹<https://www.jbl.com/bluetooth-speakers/JBL+FLIP+5-.html>

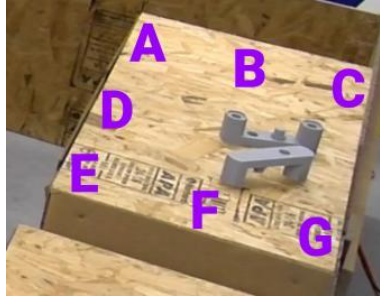


Figure 45: Photo shown to participants to answer where the robot picked. The correct answer is “F”.

physical movements.

The videos for all conditions are available on the authors’ website³⁰. In the survey, however, the videos were shown as embedded YouTube videos to avoid excessive buffering for participants from other continents, particularly Europe.

6.3.3 Questionnaire

To test the hypotheses, we asked participants to fill out a questionnaire that consists of 7-point Likert-scale, free-form, and forced-choice questions. The questionnaire allowed us to measure four subscales on causality inference, five subscales on task workload, adapted from the NASA Task Load Index, as well as four subscales on trust. After each inference and timing question, participants were asked “How confident is your answer to the question above?”

Manipulation Causality Inference & Confidence – “Where was the large wood chip that the robot tried to grasp before?” Figure 45 was shown. Options are “I don’t know” and seven choices from A to G. The correct answer is “F”.

Timing of Manipulation Causality Inference – “When did you know the answer to the question “Where was the large wood chip that the robot tried to grasp?”” The choices are: “I never knew”, “Before its head started moving around”, “While its head was moving around”, “After its head stopped moving”, “Before its arm started moving”, “When its arm started moving”, “When

³⁰<https://cs.uml.edu/~zhan/rvt/>

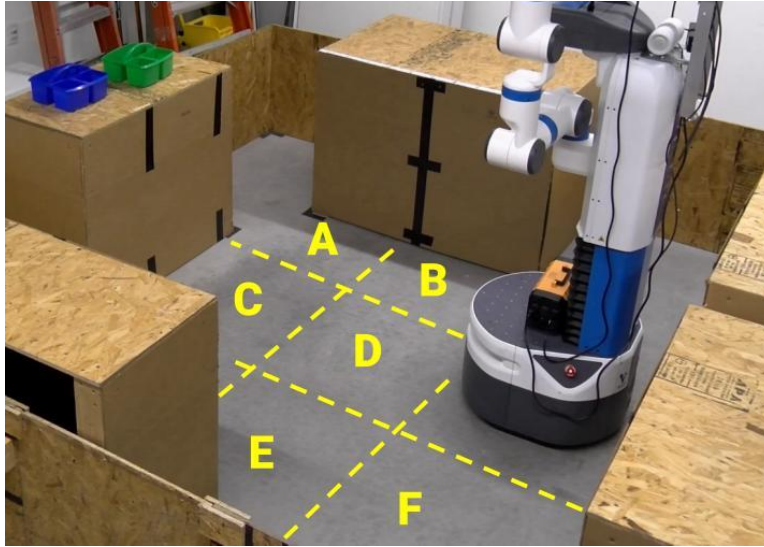


Figure 46: To answer where the ground obstacle was, this photo was shown to participants. The correct answer is “Area D”.

its hand was over the table”, “When its hand was very close to the table before grasping”, “When it was grasping”, and “Other (Please elaborate)”.

Navigation Causality Inference – “Which grid section was most occupied by the obstacle that the robot was trying to avoid before?” Figure 46 was shown. The options are “I don’t know”, “Area A”, “Area B”, “Area C”, “Area D”, “Area E”, and “Area F”. The correct answer is “Area D”.

Timing of Navigation Causality Inference – “When did you know where the obstacle was?” The choices are “Before the robot started moving”, “When the robot was facing area A”, “While the robot was moving towards the caddy table in grid E”, “While the robot was moving towards the caddy table in grid F”, “While the robot was moving towards the caddy table in grid C”, “When the robot was in front of the caddy table”, and “Other (Please elaborate)”. The correct answer is “Area D”.

Placement Causality Inference – “Which section of the caddy did the robot put the gearbox bottom into before?” A caddy photo with a labeled compartment is shown in Figure 47. Response choices include “Section A”, “Section B”, “Section C”, and “I don’t know”. The correct answer is “Section A”.



Figure 47: To answer which section of caddy the robot placed into, this photo was shown to participants. The correct answer is “Section A”.

Timing for Placement Causality Inference – “When did you know where the robot put the gearbox bottom into?” The options are “I never knew”, “Before its head started moving around”, “While its head was moving around”, “After its head stopped moving”, “When it started moving its arm”, “When its hand was over the caddy”, “When its hand was very close to the caddy before releasing the gearbox bottom”, “When it was releasing the gearbox bottom”, and “Other (Please elaborate)”.

Task Load Measures from NASA Task Load Index – We adapted the NASA-Task Load Index (NASA-TLX) [86] multidimensional scale to estimate workload for the inference tasks, which could be demanding cognitively. Specifically, we adopt the subscales of mental demand, physical demand, effort, performance, and frustration level. We decided to remove the physical demand as this study was conducted virtually and requires little physical activity; participants were allowed to finish the whole study and watch videos in a hassle-free manner at their own pace. The subscales and the options to these questions are listed below:

- *Mental Demand* – “How much mental and perceptual activity was required to answer questions after watching the videos (e.g. thinking, deciding, calculating, remembering, looking, searching, etc)? Was the task easy or demanding, simple or complex, exacting or forgiving?” The options range from very low to very high.

- *Temporal Demand* – “How much time pressure did you feel due to the rate of pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?” The options range from very low to very high.
- *Performance* – “How successful do you think you were in answering the questions after watching each video?” The options range from very good to very poor (responses are reversed to be consistent with others).
- *Effort* – “How hard did you have to work (mentally and physically) to accomplish your level of performance?” The options range from very low to very high.
- *Frustration Level* – “How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?” The options range from very low to very high.

Muir Trust scale – We used the composite trust score by Muir [120] to test our trust hypothesis, H6.5. The Muir trust score is well-established and has been used widely in the HRI and robotics literature on the trust topic, including [48, 47, 163, 12, 164]. The subscales and their options are:

- *Predictability* – “To what extent can the robot’s behavior be predicted from moment to moment?” The options are not at all, mostly not, somewhat not, neutral, somewhat, mostly, and completely.
- *Reliability* – “To what extent can you count on the system to do its job?” The options range from very low to very high.
- *Competence* – “What degree of faith do you have that the robot will be able to cope with similar situations in the future?” The options range from very low to very high.

- *Trust* – “Overall, how much do you trust the robot?” The options range from very untrustworthy to very trustworthy.

6.3.4 Quality Assurance Questions

Finally, we asked several attention check questions to help us ensure participant attention to the experimental stimuli, similar to those used in Brooks et al. [26]. The questions are:

- After watching picking videos – What is the color of the gearbox bottoms? The choices are “Blue”, “Red”, “Green”, and “Gray”. The correct answer is “Gray”.
- After watching navigation videos – What is the color of the robot? The choices are “Mostly white”, “Mostly red”, “Mostly yellow”, and “Mostly green”. The correct answer is “Mostly white”.
- After watching placing videos – How many robot(s) were in the video? The choices are from 0 to 3. The correct answer is “1”.

The choices to these attention check questions were displayed in random order. In addition, we also added a Google reCAPTCHA³¹ verification question at the beginning of the survey to avoid bots – scripts to automate question answering.

6.3.5 Procedures

The study was conducted on Prolific, a similar platform to Amazon Mechanical Turk for online participants recruitment. Participants entered the study via an anonymous link to a Qualtrics survey³². Once started, participants were presented with informed consent information and the Google reCAPTCHA verification. After agreeing to participate and passing the verification, participants

³¹<https://www.google.com/recaptcha/about/>

³²https://umasslowell.col.qualtrics.com/jfe/form/SV_dmVGfnPCdJ1ndEG

were presented with demographic questions and randomly assigned to one of the experimental conditions.

Before watching each video, participants were presented with the motivating scenarios and the prompt questions, as seen in the Introduction section. Then they watched the videos, embedded from YouTube, and answered questions on the same page. We also gave the YouTube links that open in a new tab to foresee potential technical difficulties; the text is “If the video doesn’t load, please click this YouTube link³³. It will open in a new tab/window”.

For those conditions where the videos have sound, we first showed a YouTube video with sound only³⁴ and asked what did they hear to ensure participants can hear the sound. This video is presented on a separate webpage as we found YouTube remembers watchers’ mute preference, and this can cause other videos on the same webpage to remain muted, even this sound-only video is manually unmuted.

After watching all videos and answering relevant questions, participants are then asked to answer the trust and NASA Task Load Index questionnaire to finish the study. To record participants as complete, participants are redirected back to Prolific at the end.

The entire study took an average of 13.6 minutes to complete, with a median of 11.6 minutes. All participants are paid US \$3.01 at an hourly rate of \$9.50, estimated for 19 minutes by the experimenter before the study. The study was approved by the institutional review board (IRB) at the University of Massachusetts Lowell in the USA.

6.3.6 Power Analysis, Participants, and Participants Recruitment

We used G*Power 3.1.9.7 [59] to perform two *a priori* power analyses because we planned to run two types of hypothesis tests.

We first performed an *a priori* power analysis for “Goodness-of-fit tests: Contingency ta-

³³The YouTube link is a hyperlink.

³⁴Video to test sound: <https://www.youtube.com/watch?v=jsJsnLZeJa0>

bles”. The parameters were: Effect size $w = 0.5$ for large effect size, α error probability = 0.05, Power ($1 - \beta$ error probability) = 0.95, Df = 9 which reflected the number of fixed choices in our measures described in section 4.3. The output parameters in G*Power showed that the sample size to reach desired power $1 - \beta = 0.95$ was 84 for a single goodness-of-fit test. Thus, for the seven conditions of our experiment design, we needed at least $7 \times 95 = 665$ participants.

We also performed an *a priori* power analysis for “ANOVA: Fixed effects, omnibus, one-way” tests. The parameters were: Effect size $f = 0.4$ for large effect size, α error probability = 0.05, Power ($1 - \beta$ error probability) = 0.95, and Number of groups = 7, reflecting the number of independent conditions in our study. The output parameters showed that the total sample size needed was 140.

Thus our study would need approximately $N = 665$ participants to be sufficiently powered for both types of statistical tests.

Using Prolific, we recruited a total of 691 participants, with only 25 (3.6%) of them failed the quality check questions. The randomizer feature in Qualtrics is used to ensure evenly presented condition assignment. This resulted in 666 valid cases with one extra participant in the Replay-Projection condition, which might be caused by timed-out participants taken one of the other conditions. To ensure that we had an equal number of participants in each of the seven conditions, we trimmed the data from the extra participant. This procedure resulted in a sample size of $N = 665$ with 95 participants in each of the seven between-subjects conditions.

The final sample of 665 participants includes 267 females, 391 males, 5 non-binary, and 1 transgender person. Their age ranges from 19 – 88, $M = 31.8$, *median* = 28.0.

Specified qualifications for participation on Prolific included being over 18 years old, fluent in English, which provided a reasonable assumption of English language comprehension, having taking part in 100 – 10,000 studies on Prolific, and a 100% approval rating³⁵. Each participant, whether or not they passed data quality assurance checks, was paid for their participation, although

³⁵Prolific uses the upper bound of the 95% confidence interval to calculate approval rate.

as noted above, the data for people who failed data quality assurance checks was removed from our analysis.

6.3.7 Implementation

The action sequence in the replay is implemented using ROS and Behavior Trees [40]. For more details on how we modeled the mobile kitting task in Behavior Trees, please see [82].

For the replay, we recorded relevant ROS topics to a MongoDB database for its schemaless feature (no need to create a table for each ROS topic data type) and querying capabilities, including arm movement, neck, and eye camera movement, as well as wheel movement. At replay time, these topics are queried from the database and streamed back to ROS to exactly replicate the movement that happened at record time.

As projection markers are only used to show perception and manipulation intents, the timing choice is rather simple and static. The robot is programmed to project right after objects are recognized and the object target to be grasped is determined.

To activate verbal markers, we used the task-relevant heuristics to maximize influences to the attention process, which is one of four serial processes that observational learning depends on [16]. These heuristics are informed by research on instructional features [140] in observational learning and are commonly known as learning points (also referred as codes) [154].

The heuristic used for picking and placing is to speak when the robot's gripper is above the manipulation target to both attract attention and avoid ambiguity. For navigation, the robot speaks right after a navigation plan is computed and its base is about to start moving. All the robot speeches are listed in Table 7.

6.4 Results

We used R to analyze the data. As we have seven conditions and a considerable number of choices in our multiple-choice questions, we decided to annotate relevant figures with pairwise statistical

test results, including the p values and significance levels. One to four asterisks (*, **, ***, and ****) indicate $p < 0.05$, $p < 0.01$, $p < 0.001$, and $p < 0.0001$ respectively. The abbreviation of *n.s.* denotes “not significant” statistically. When discussing pairwise results, the statistics included are all of statistical significance; any non-significant results discussed are explicitly mentioned as such.

For 7-item Likert responses, we coded them -3 to 3, from the least to the greatest extent. For example, “very unsure” would be -3 while “very confident” is coded 3.

6.4.1 H6.1. Effective causal inference with verbal markers (partial support)

To see whether verbal markers are effective for aiding participants in causal inference, we first analyze the responses to the inference questions for picking, navigating, and placing subtasks.

For all inference types, including picking, navigation, and placing, we were able to find statistically significant results from proportions tests. We first ran chi-square goodness-of-fit tests on the conditions and responses to all the multiple-choice inference questions, which reveals statistical significance ($p < 0.0001$) for each type of inference. Post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons were performed and revealed significant differences for each type as well.

For the **picking inference** responses in Figure 48, there are statistically significant differences in all responses across all *replay* conditions (the top four subfigures), where almost all participants correctly inferred that F is where the robot has picked.

In the Say condition, statistically significant differences were found for choices A, C, E, and G. Around half of the participants (47, 49.5%) selected the nearby E, while the correct answer, F, was not of significance. For the other two non-replay conditions, Project and Project-Say, around half of participants have the correct inference (48, 50.5% participants for Project and 53, 55.8% for Project-Say).

The results suggest that, without the physical replay of the head and arm movements, par-

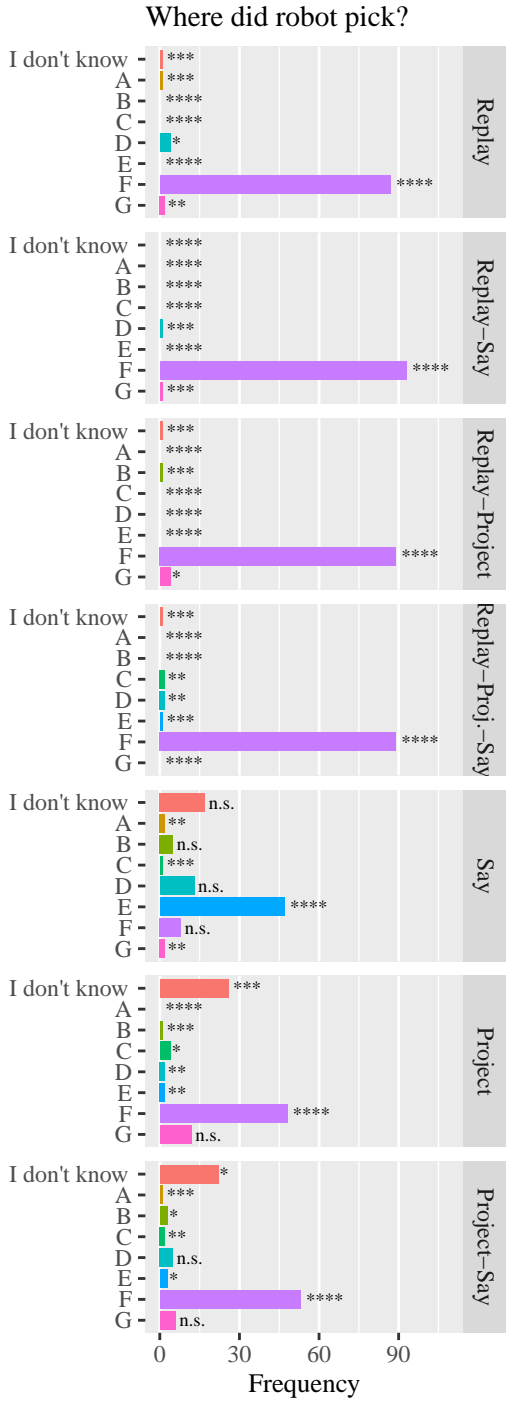


Figure 48: Manipulation inference responses. “F” is correct. Replay conditions perform the best: nearly all participants were correct. Half wrongly selected nearby E in Say. Project and Project-Say only have half participants correct.

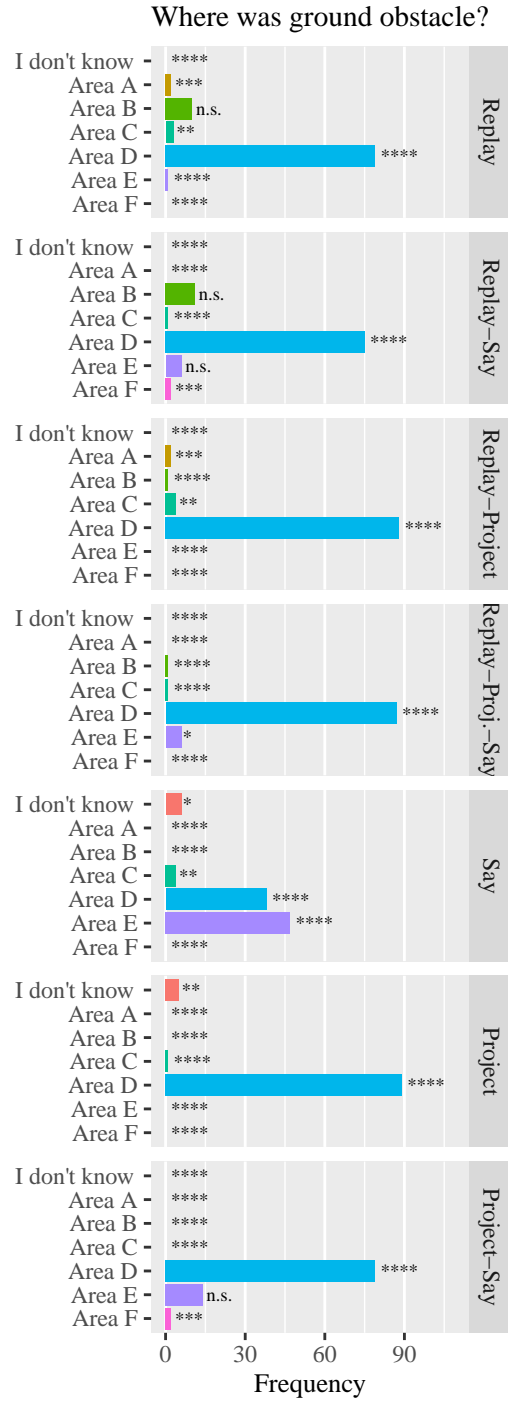


Figure 49: Navigation inference responses. The correct answer is “Area D”. Most participants were correct in all conditions except for the Say condition, in which only 40% were correct and half selected nearby Area E.

participants had difficulties in the picking inference. Particularly, with verbal indications alone, participants had an even harder time inferring the correct picking location, F, because they chose the nearby E.

For the **navigation inference** responses in Figure 49, there are statistically significant differences in almost all responses across all conditions. Except for the Say condition, 78% - 93% participants were able to infer the correct answer that Area D was where the ground obstacle was (Replay: 79 participants, 83%; Replay-Say: 75, 78.9%; Replay-Project: 88, 92.6%; Replay-Proj.-Say: 87, 91.6%; Project: 89, 93.7%; Project-Say: 79, 83.2%). For the Say condition, only 40.0% of participants were able to infer correctly, while 49.5% of participants chose the nearby E.

Similar to the picking inference, half of the participants may have interpreted right as far-right, which Area E is, in the robot's indication for the ground obstacle location.

For the **placement inference** responses in Figure 50, we were able to find statistically significant results in almost all choices (except for Section B) across almost all conditions except for the Project condition. Around 60% of participants were able to infer correctly in non-Project conditions: Replay: 56 participants, 58.9%; Replay-Say: 57, 60.0%; Replay-Project: 57, 60.0%; Replay-Proj.-Say: 64, 67.4%; Say: 54, 56.8%; Project-Say: 58, 61.1%.

The responses of Section B may happen by chance in all conditions except for the Replay-Say condition, in which there is only weak support by a $p < 0.05$ significance.

In the Project condition, unfortunately, all choice responses may happen by chance (*n.s.*).

The placement inference results suggest that the effectiveness of verbal indications is supported.

In conclusion, **H6.1** is partially supported. Participants were not able to infer the location of both picking and the ground obstacle with verbal markers. They can only infer where the robot placed the gearbox bottom.

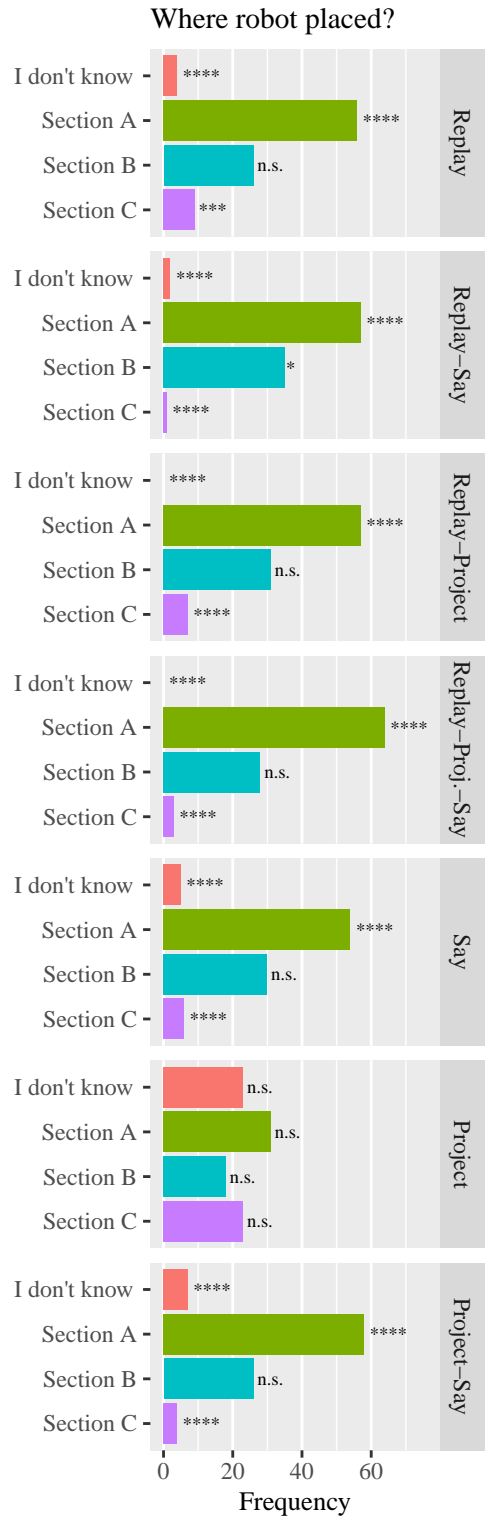


Figure 50: Placement inference responses. “Section A” is the correct answer. Around 60% participants could infer correctly except for Project, in which no statistically significant results were found.

6.4.2 H6.2. Effective causal inference with projection markers (partial support)

As we explore the answer to H6.1, we can also test whether projection markers are effective.

For picking inference, shown in Figure 48, the project indications alone (Project) and the one with verbal indications (Project-Say) are only 50% effective compared with those conditions with physical replay. However, like what we have discussed H6.1, projection indicators are more effective than verbal indicators when they are the only cues being presented by the robot, because half participants were wrong in the verbal-only condition of Say.

In terms of navigation inference, shown in Figure 49, all conditions with projection indications (Replay-Project, Project, Project-Say) are remarkably effective – the majority of participants were able to infer the location of the ground obstacle. Compared to verbal indicators alone, projection indicators alone are one time more effective.

Regarding placement inference, shown in Figure 50, projection indicators with either physical replay (Replay-Project) or verbal indicators (Project-Say) are at least the same as other non-projection conditions in terms of effectiveness. However, all the responses to the navigation inference questions under the Project condition may happen by chance (*n.s.*), so we are not able to conclude the effectiveness of projection indicators alone.

Thus, **H6.2** is partially supported. Conditions with projection indicators are remarkably effective for inferring the location of ground obstacles but are only half effective for picking inference when the projection is the only indicator present. With insignificant results in the placement inference responses from Project, their effectiveness remains unknown.

6.4.3 H3 – Efficiency. Faster causal inference with projection markers (partial support)

To check whether projection markers lead to faster causal inference, we analyzed participants' responses to the timing questions. We did not have this type of question for the Project condition in the navigation because ground projection was always on throughout the inference video, thus

no early or later events were present. However, we can still gain insight into whether projection indicators alone had an effect when analyzing the responses from participants who experienced the Project-Say condition in its navigation video.

For **picking inference timing**, we first conducted a chi-square goodness-of-fit test on the responses to the picking inference timing question across all conditions, and it reveals a statistically significant result ($p < 0.0001$). Then we ran post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons, the results are annotated on Figure 51.

Comparing the Project condition with the replay conditions, approximately the same number of participants (~ 35 , 36.8%) in the Project condition chose “After its head stopped moving” (in green) as the number of participants in replay conditions chose “When its hand was very close to the table before grasping” (in violet).

While this means the Project condition accelerates picking inference, 40 participants (42.1%, the most in all conditions) reported that they never knew the answer in the Project condition. In general, this happens in all non-replay conditions, as seen in the red bars of the bottom three subgraphs in Figure 51: 21 participants (22.1%) in Say, 40 (42.1%) in Project, and 27 (28.4%) in Project-Say selected “I never knew”. They are all of statistical significance.

We also analyzed the conditions where the projection indicators are accompanied by replay and verbal indicators.

Adding replay (Replay-Project) or both replay and verbal indications (Replay-Project-Say) also accelerates picking inference. Seen from the purple and more saturated blue bars in the third and fourth replay subfigures (Replay-Project and Replay-Project-Say) in Figure 51, the number of participants who inferred “when it was grasping” decreased compared to the Replay and Replay-Say conditions that do not have any projection indicators (top two subfigures in Figure 51). And the earlier event “When its hand was over the table” has more participants than the Replay and Replay-Say conditions. The statistics for those two events are both of significant difference.

However, when adding verbal indicators to projection (Project-Say) or in the Say condition,

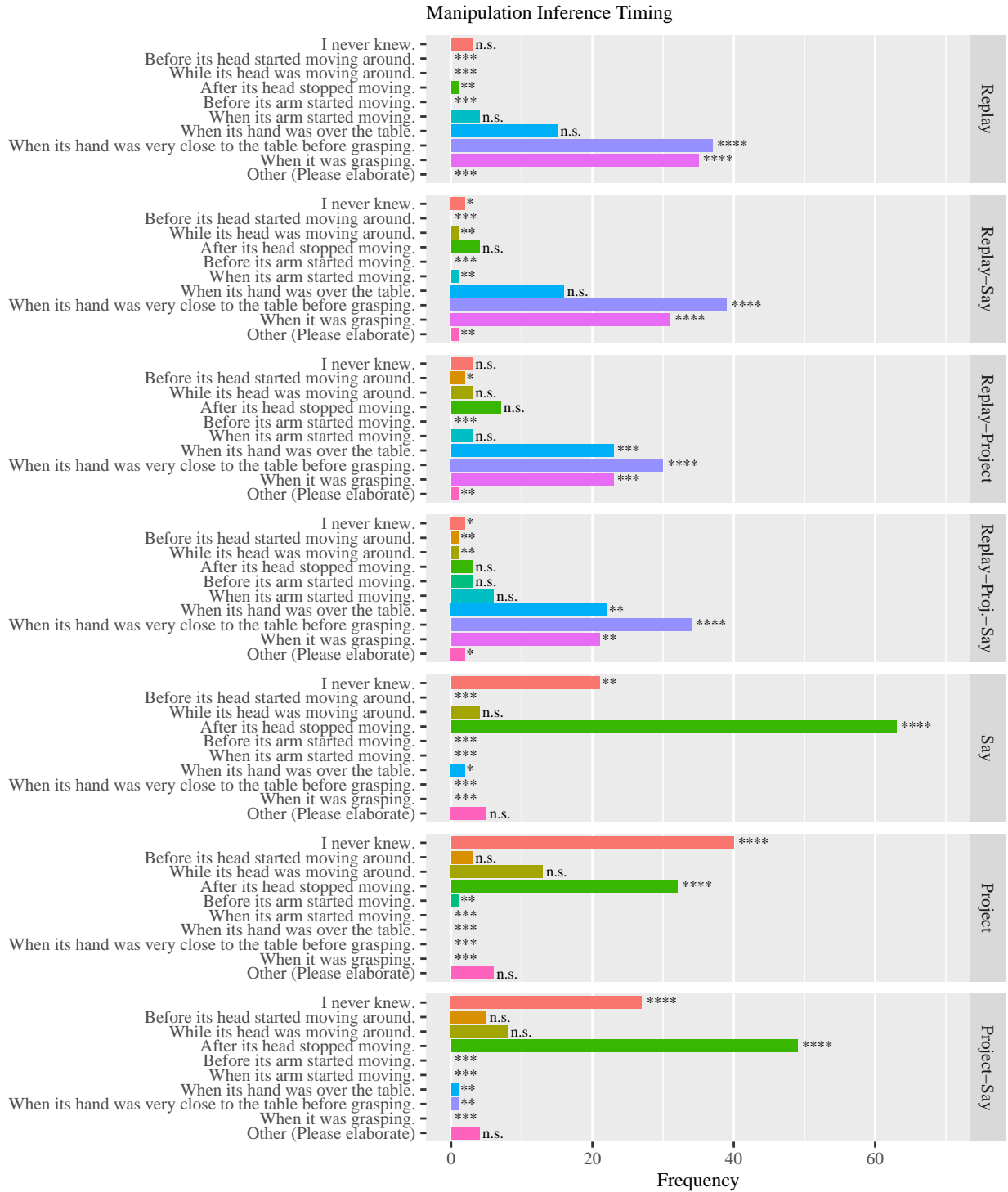


Figure 51: Participant’s responses to when they have inferred the picking location. The Say condition performs the best with 60+ correct participants but 20+ never know. Project and Project-Say are at the second tier with fewer correct participants and more unknowing participants. In all replay conditions, the top four in the figure, participants reported they know at a later event.

fewer participants chose “I never knew”. Instead, 32.6%, 31 more participants (66.3% vs. 33.7%, 63 vs. 32) in the Say condition and 17.9%, 17 more participants (51.6% vs. 33.7%, 49 vs. 32)% in the Project-Say condition inferred the picking location after “After its head stopped moving” (See the green bars in Figure 51).

Thus, we can conclude that adding projection markers to physical replay makes participants’ picking inference faster, but not when included in verbal indicators. In terms of the maximum number of participants who infer early, the verbal condition performs the best but many participants (21, 22.1%) reported that they never know where the robot picked. To ensure almost all participants infer the picking location, projection with replay (Replay-Project) and projection with both replay and verbal indicators (Replay-Project-Say) are two better choices that are approximately the same.

For **navigation inference timing**, we conducted the same type of chi-square goodness-of-fit test on the responses to the navigation timing questions in all conditions except for the Project condition where timing is irrelevant because the robot projects throughout the video. Post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons are conducted when there is a statistically significant difference revealed by the goodness-of-fit tests.

For replay conditions where the robot’s base moved, the chi-square goodness-of-fit test shows a statistically significant difference across responses ($p < 0.0001$). Pairwise comparison results are annotated on Figure 52.

As seen from the figure, around half of participants who experienced the Replay and Replay-Say conditions (top two subfigures), 47 (49.5%) in Replay and 55 (57.9%) in Replay-Say, were able to infer the ground obstacle location while the robot was in grid E (during left rotation, before leaving for F; fourth cyan bars in Figure 52).

When projection indicators are added, the numbers dropped half to 28 (29.5%) in both Replay-Project and Replay-Project-Say conditions. Instead, 26 (27.4%) participants, approximately the same as the dropped number, inferred the ground obstacle location at an *earlier* event of “before the robot started moving” in the Replay-Project-Say condition (the first bar in the last

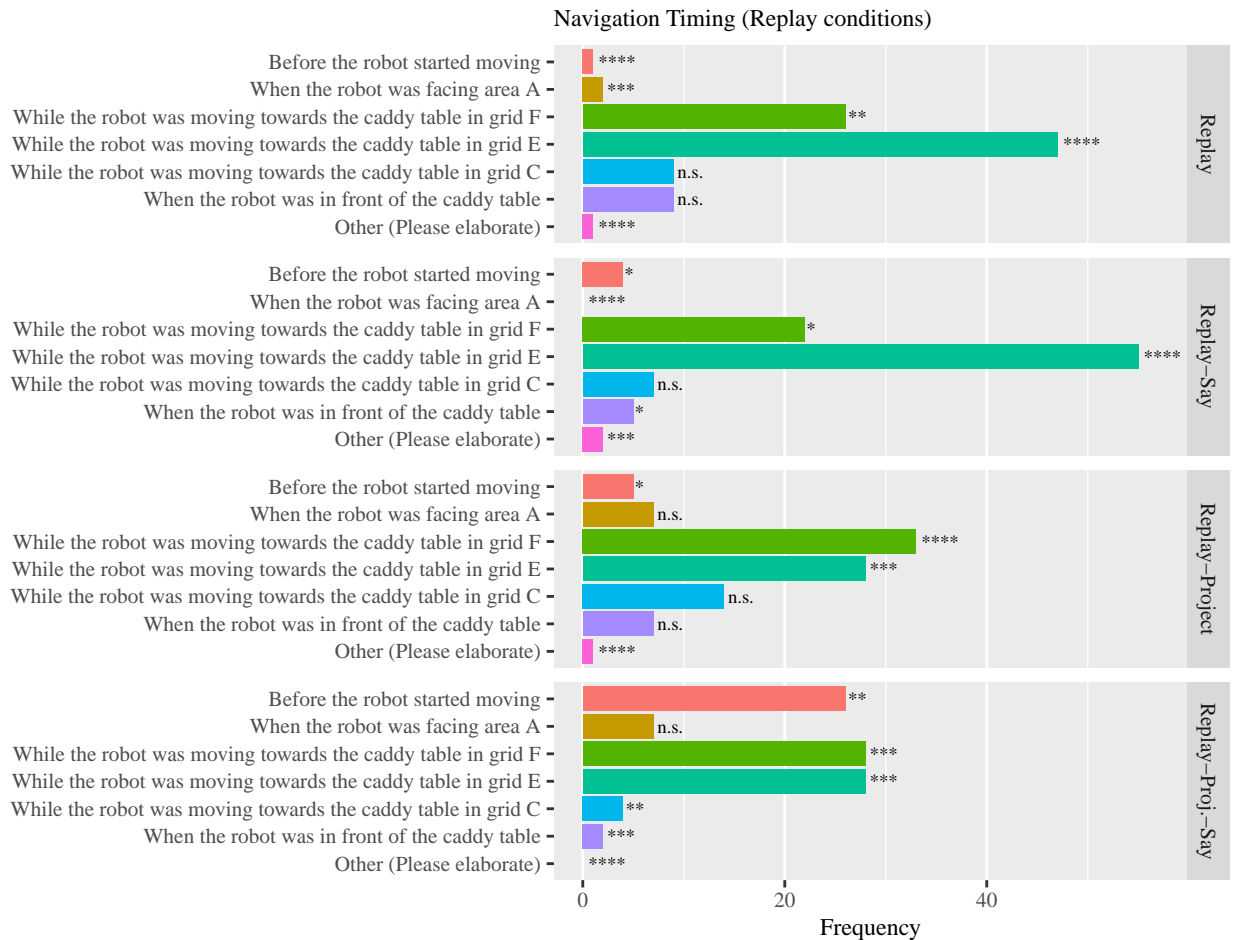


Figure 52: Participant’s responses to when they have inferred where the ground obstacle was in replay conditions. In summary, Replay-Project-Say performed the best, with 20+ participants inferring at the earliest event: before the robot started moving. (Non-replay conditions, excluding the Project condition where projection was always on, had their own options as the robot’s base did not move during these conditions; See Figure 53 and 54.)

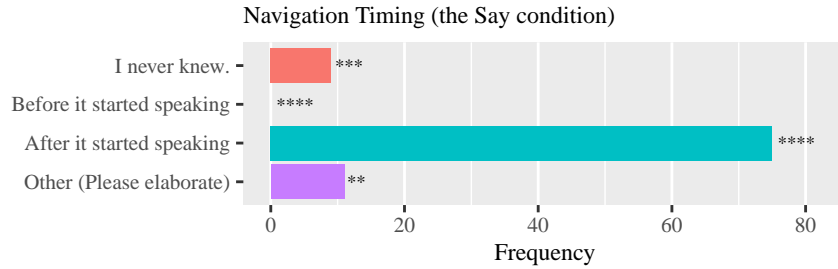


Figure 53: Participant’s responses in the Say condition to when they have inferred where the ground obstacle was. Most participants made the inference after the robot started speaking, more than any option in the replay conditions shown in the previous figure (Figure 52). (The options were only present in the Say condition as the robot did not move its base but just spoke.)

subfigure of Figure 52). For the Replay-Project condition, the dropped number of participants are distributed to “when the robot was facing area A” and a later event of “while the robot was moving towards the caddy table in grid C”, but unfortunately post-hoc binomial tests suggest that they may happen by chance.

In the Say condition, binomial tests with Holm-Bonferroni correction suggest a statistically significant difference across all options. Results are shown in Figure 53. 78.9% participants (75) were able to make the ground obstacle inference “after it started speaking”. No participants reported that they made the inference “before it started speaking”.

The result here is consistent with the immediately previous finding that Replay-Project-Say accelerates the participants’ inference as the results suggest that verbal indicator is remarkably effective.

In the Project-Say condition, we performed the same statistical test as in the Say condition and results are shown in Figure 54. Sixty participants (63.2%) were able to know where the ground obstacle was “before it started speaking (At the beginning of the video, with projection)”. While only 30 participants (31.6%) reported they know it “after it started speaking“, the binomial tests with Holm-Bonferroni correction, unfortunately, suggest that this might just happen by chance. Nonetheless, comparing responses in the Say conditions to those in the Project-Say condition, more than 60% more participants could make the inference with projection alone.

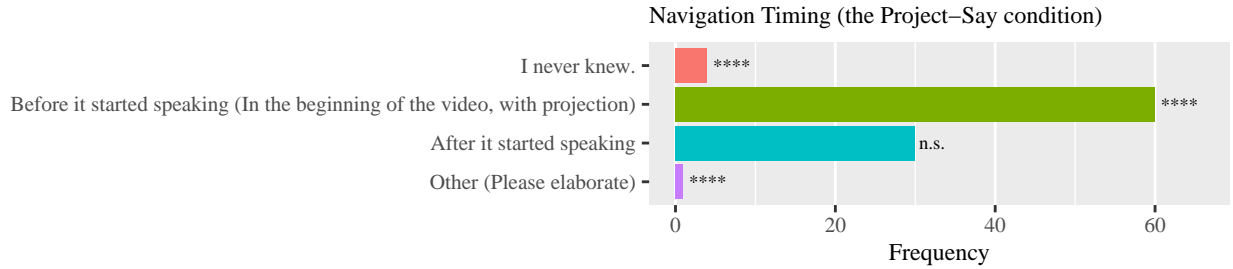


Figure 54: Participant’s responses to when they have inferred where the ground obstacle was. More than 60% of participants made the inference from the ground projection. (The options were only present to the Project-Say condition as the robot did not move its base but made projection onto the ground and spoke.)

Thus, we reach a mixed conclusion again, similar to the one for picking timing inference. Including projection indications in replay condition, whether with verbal indications, defers the inference from staying at its place (“While the robot was moving towards the caddy table in grid E”) to leaving for the caddy (“While the robot was moving towards the caddy table in grid F”). With projection indications only, more than 60% of the participants were able to make the inference from the projection before it started speaking, which is suggested by analyzing the responses in the Project-Say condition.

Finally, we analyzed the responses to the **placement inference timing** questions. The same as other analyses in this section, we ran a chi-square goodness-of-fit test and it revealed a statistically significant difference ($p < 0.0001$). Results from post-hoc binomial tests with Holm-Bonferroni correction for pairwise comparisons are shown in Figure 55.

Again, the Project condition has the most participants (50, 52.6%) who were able to infer the placement position “After its head stopped moving” (the green bar in the second last subfigure of Figure 55) while only 36 and 40 participants (37.9% and 42.1%) in the Say and Project-Say conditions were able to do so after the same event. For replay conditions, fewer than eight participants in all statistically significant events that happen before its hand over the caddy were able to make the inference.

However, similar to the responses to the picking inference question, the Project condition

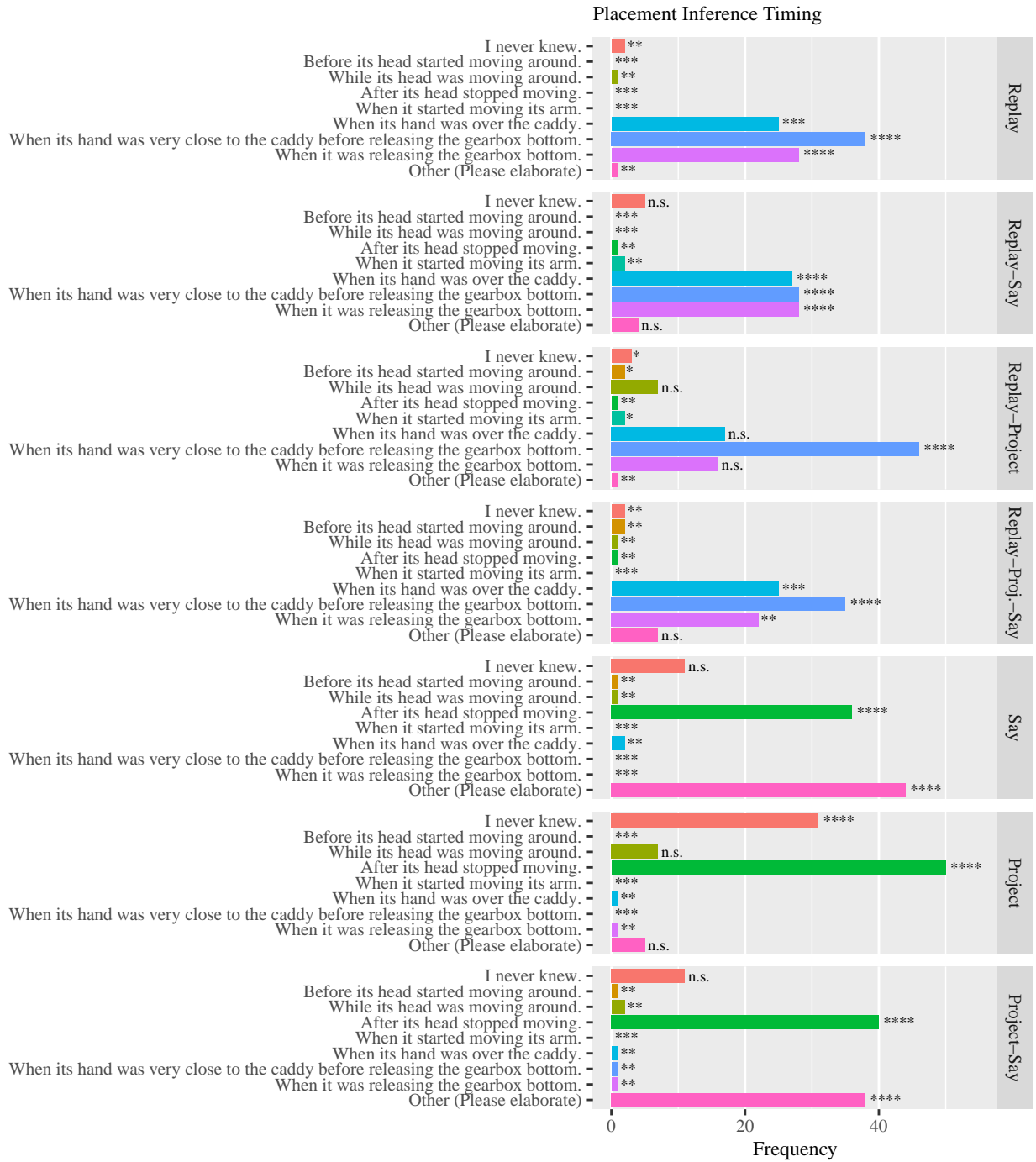


Figure 55: Participant’s responses to when they have inferred which section of the caddy the object was placed into. Participants in the Project condition made the earliest inference after the robot’s head stopped moving. However, 30+ participants reported they never knew. Say and Project-Say has the top performance because the participants elaborated on what they knew from the robot’s speech. For replay conditions, arm movement significantly delays the inference.

had the most participants (31, 32.6%) who chose “I never knew” (pink bar). Only a few participants chose this option in the Replay, Replay-Project, and Replay-Project-Say conditions. The responses to this option in other conditions, i.e. Replay-Say, Say, and Project-Say, are not statistically significant and may have just happened by chance.

For replay conditions, most responses are distributed in three late events when the robot’s gripper is above the caddy to get closer and release the object: “When its hand was over the caddy”, “When its hand was very close to the caddy before releasing the gearbox bottom”, and “When it was releasing the gearbox bottom”.

It is worth mentioning that for the Say and Projection-Say conditions, 44 and 38 participants (46.3% and 40.0%) chose to elaborate on a different choice. Upon analyzing these responses, all participants said they knew when the robot said so, which is the same event as “After its head stopped moving”. Therefore, the Say and Project-Say are the fastest conditions for participants to infer the placement position. Compared to the replay conditions that we just discussed, the robot’s physical arm movement delayed the inference.

Thus, the efficiency aspect of projection indicators in **H6.3** is partially supported. Projection indicators performed the best to help participants infer the ground obstacle as early as the obstacle is indicated by projection. However, for inferring picking and placing locations, while projection expedites the inference, it also prevents one-third of participants from making any inference (30 to 40 participants respectively – 31.6% to 41.1%).

6.4.4 H3 – Accuracy. More accurate causal inference with projection markers (not supported)

To test whether projection markers make causal inference more accurate, we analyzed the nonparametric responses to participants’ confidence ratings in their answers to all three inference questions. More accurate methods should make participants more confident.

For participants’ confidence in their **picking** inference responses (See Figure 56), we first

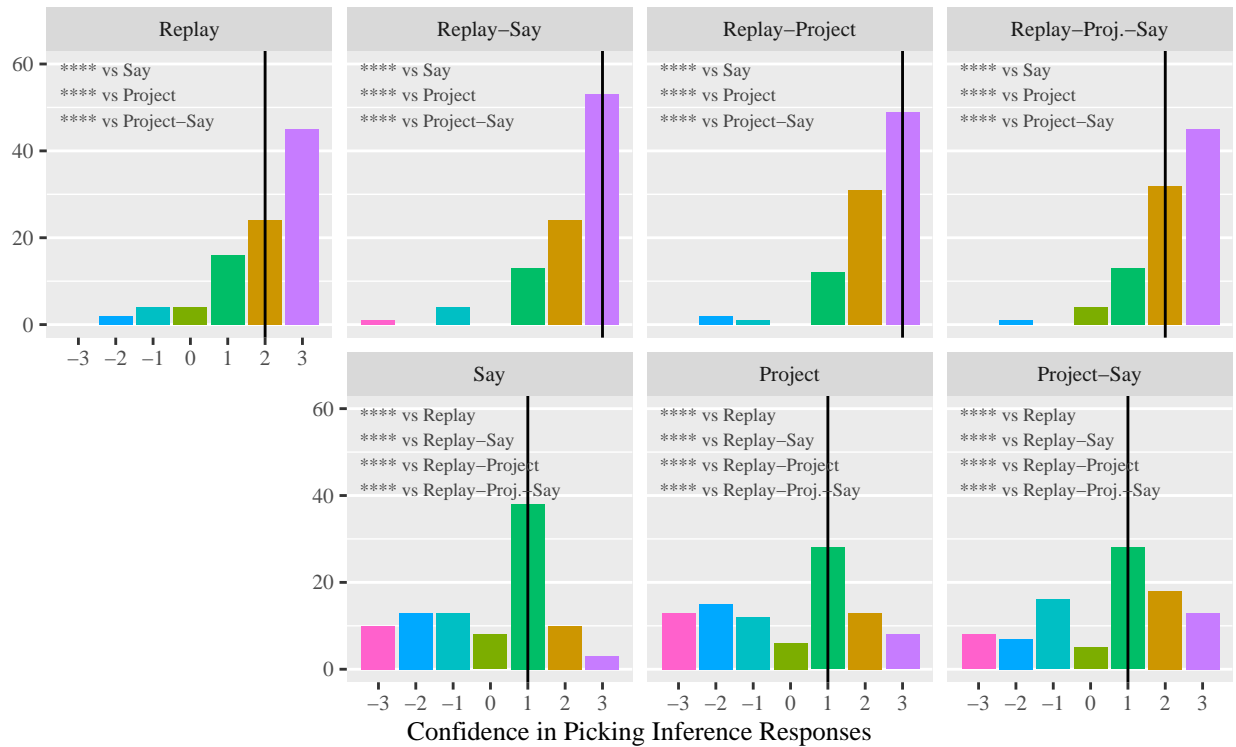


Figure 56: Participant's confidence levels in their responses to the picking inference question. Generally, participants in replay conditions are more confident than those in non-replay conditions (Confident or very confident in replay conditions vs. somewhat confident in non-replay conditions).

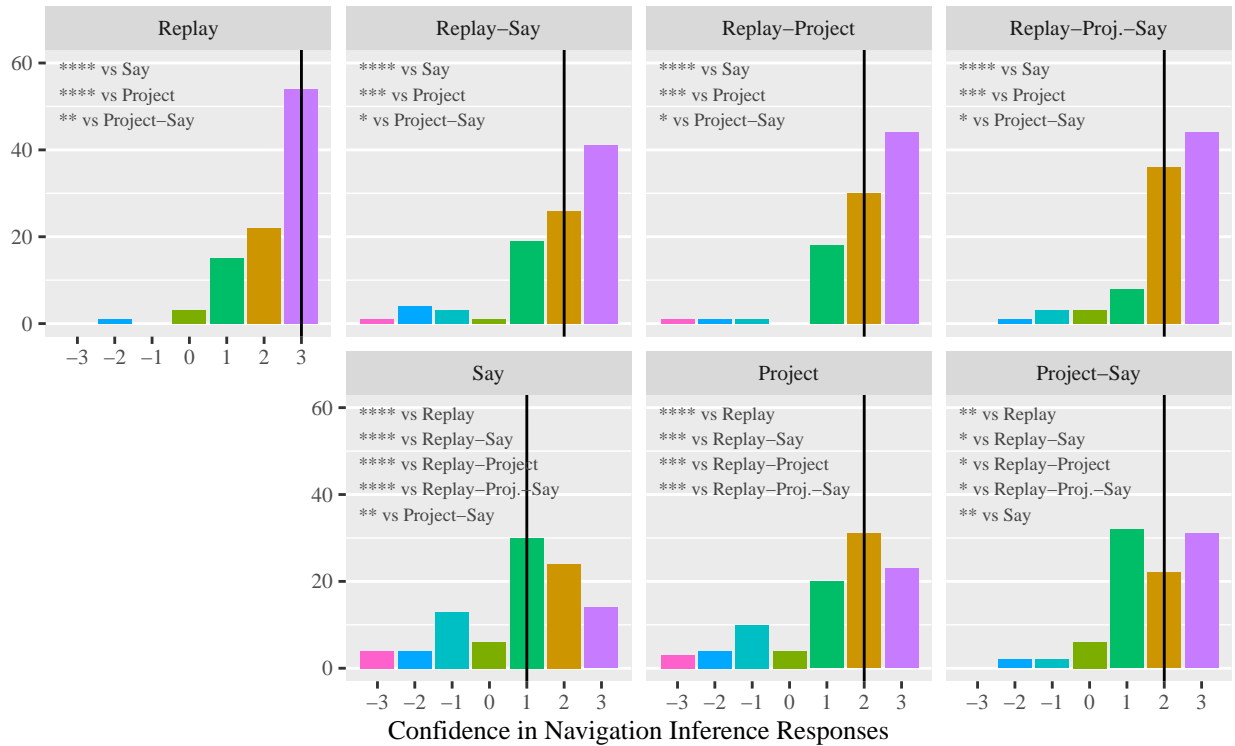


Figure 57: Participant’s confidence levels in their responses to the navigation inference question. Participants’ confidence levels in the Project and Project-Say conditions are increased to confident from somewhat confident in their picking inference. However, the statistical significances suggest that participants are still more confident in replay conditions (more right-skewed).

conducted a Kruskal-Wallis H test and it reveals a statistically significant difference across conditions ($\chi^2(6) = 240.02, p < 0.0001$). We then run post-hoc Mann-Whitney U pairwise comparisons with Holm-Bonferroni correction. Results show there are significant differences between the Replay condition and the conditions of Say, Project, and Project-Say (all: $p < 0.0001$), between Replay-Say and the conditions of Say, Project, and Project-Say (all: $p < 0.0001$), between Replay-Project and the conditions of Say, Project, and Project-Say (all: $p < 0.0001$), and between Replay-Project-Say and the conditions of Say, Project, and Project-Say (all: $p < 0.0001$).

The results show that participants in replay conditions are more confident in their responses than non-replay conditions, i.e., very confident in replay conditions vs. somewhat confident in non-replay conditions.

For participants' confidence in their **navigation** inference responses (Figure 57), a Kruskal-Wallis H test shows that there is a statistical significant difference across conditions ($\chi^2(6) = 79.125, p < 0.0001$). Post-hoc Mann-Whitney U pairwise comparisons with Holm-Bonferroni correction show the statistically significant results between the Replay condition and the conditions of Say ($p < 0.0001$), Project ($p < 0.0001$), and Project-Say ($p < 0.01$), between Replay-Say and the conditions of Say ($p < 0.0001$), Project ($p < 0.001$), and Project-Say ($p < 0.05$), between Replay-Project and the conditions of Say ($p < 0.0001$), Project ($p < 0.001$), and Project-Say ($p < 0.05$), between Replay-Project-Say and the conditions of Say ($p < 0.0001$), Project ($p < 0.001$), and Project-Say ($p < 0.05$), and between Say and Project-Say ($p < 0.01$).

These results are statistically consistent with the data of picking inference confidence, plus a statistical significance between Say and Project-Say. It is worth noting that participants' confidence raised to a median of confidence in Project and Project-Say from somewhat confident in picking reference (compare the last two subfigures in Figure 57 with the last two in Figure 56). Although the median value is increased, the statistically significant results suggest that participants in replay conditions are still more confident, as seen to be more right-skewed in the left four subfigures in Figure 57.

For participants' confidence in their **placement** inference responses (Figure 58), unsurprisingly, Kruskal-Wallis H test shows a statistical significant difference across conditions ($\chi^2(6) = 53.436, p < 0.0001$). Post-hoc Mann-Whitney U pairwise comparisons with Holm-Bonferroni correction show the statistically significant results between the Replay condition and the conditions of Replay-Project-Say ($p < 0.05$), Project ($p < 0.01$), between Replay-Say and the conditions of Replay-Project ($p < 0.05$) and Project ($p < 0.0001$), between Replay-Project and the conditions of Replay-Project-Say (0.01) and Project ($p < 0.05$), between Replay-Project-Say and the conditions of Say ($p < 0.05$) and Project ($p < 0.0001$), and between Say and Project ($p < 0.05$). For the Project-Say condition, no statistically significant results were found.

These suggest that participants who experienced Project and Replay-Project projection meth-

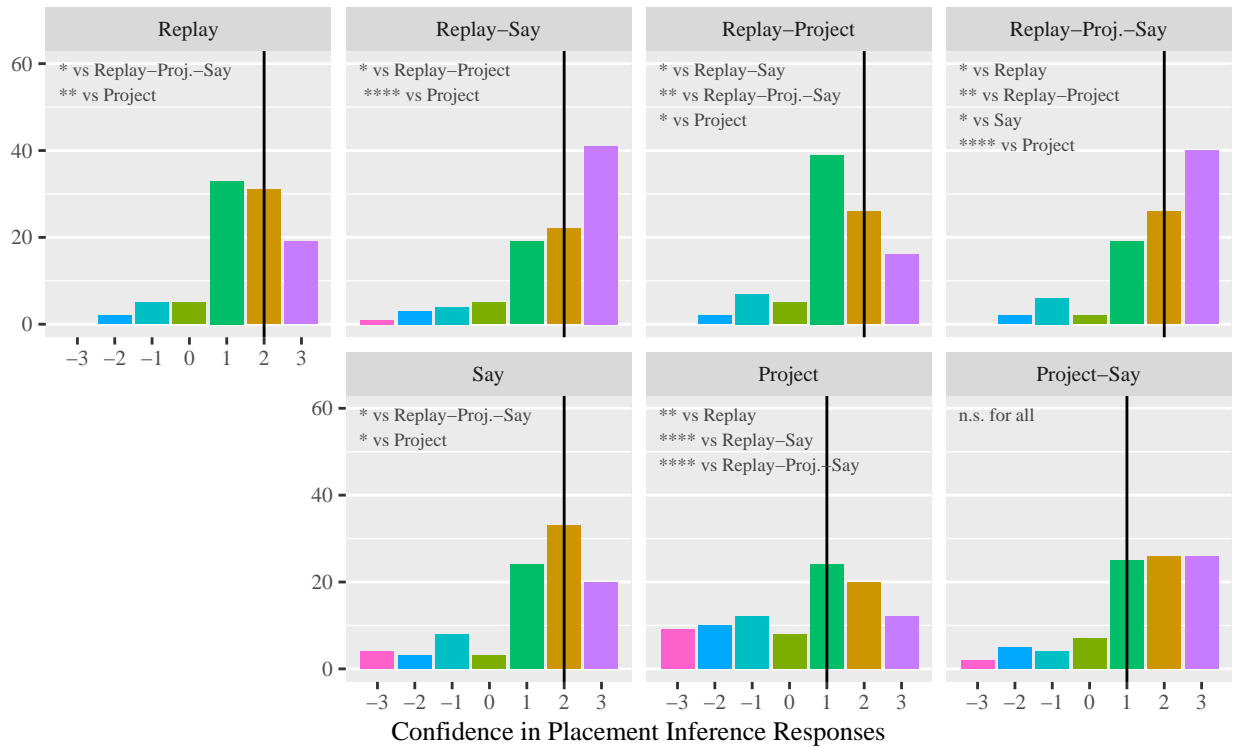


Figure 58: Participant’s confidence levels in their responses to the placement inference question. Participants in the Project condition has more unsure ratings, while other condition has more participants distributed in different confidence level ratings.

ods are less confident than other non-projection replay conditions, i.e., somewhat confident vs. confident. It is worth noting that after adding verbal indicators to Replay-Project (Replay-Project-Say), participants become more confident. The Say condition itself and Project-Say also perform as well as the first three replay conditions because there is no statistical significance between each pair.

Thus, the accuracy aspect of **H6.3** may not be supported through the confidence measure. With projection alone, it is not more accurate than other replay conditions because people are less confident in their inferences. Replay conditions perform well in almost all subtasks, while verbal indicators make participants as confident as replay conditions in placement inference.

6.4.5 H6.4. The same workload in both verbal and projection conditions (mostly supported)

To investigate whether participants bear the same workload in both verbal and projection conditions, we analyzed the Likert responses to the NASA Task Load Index questionnaire. Figure 59 shows a bar chart that visualizes the data.

We ran Kruskal-Wallis H tests across the subscales, which reveal statistical significant differences for all: mental demand ($\chi^2(6) = 42.3, p < 0.0001$), temporal Demand ($\chi^2(6) = 15.4, p < 0.05$), performance ($\chi^2(6) = 176, p < 0.0001$), effort ($\chi^2(6) = 31.1, p < 0.0001$), and frustration Level ($\chi^2(6) = 93.5, p < 0.0001$). We then ran post-hoc Mann-Whitney U pairwise comparisons with Holm-Bonferroni correction and the results are in Figure 59.

Surprisingly, projection indicators alone had a median rating of somewhat high (coded as 1) in the need of mental demand. Statistically significant differences were found between the Project condition and each of the replay conditions (See column 1, row 6 in Figure 59). No significant differences were found between the Projection condition and the Say and Project-Say conditions.

For temporal demand, every condition was rated low to some extent: either somewhat low (coded as -1, in Project and Project-Say) or low (coded as -2, in all replay conditions and Say). No statistically significant differences between the Project condition and all other conditions were

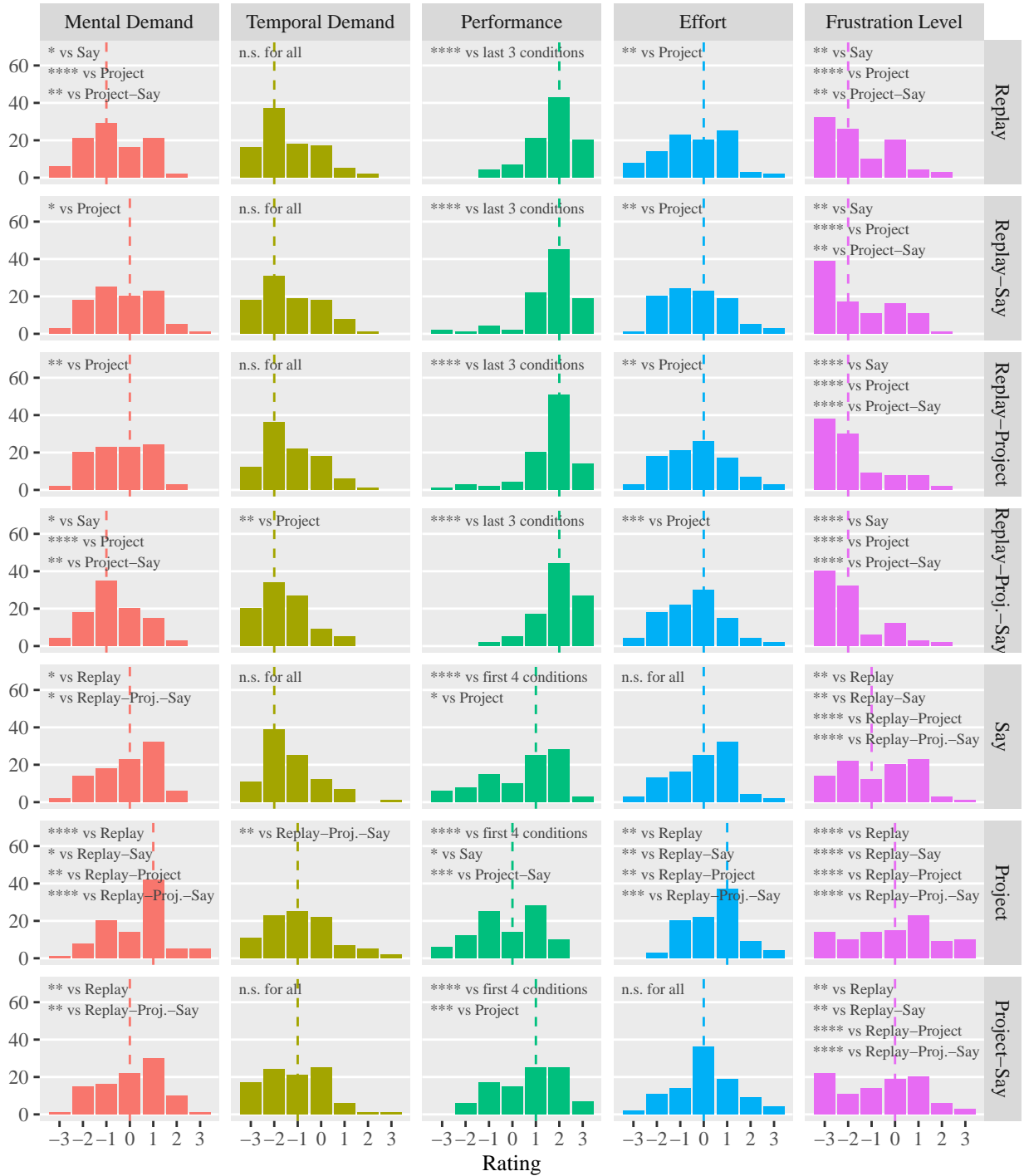


Figure 59: Responses to the NASA Task Load Index questionnaire. Results from pairwise comparisons are shown and dashed lines indicate median values. In general, replay conditions and the Say condition performed the best; No statistically significant differences were found between the Say and Project conditions except for performance. See Section 6.4.5 for more details.

found except for the Replay-Project-Say condition.

In terms of participants' performance, the Project condition is the only case not rated good to some extent – the median rating is neither good nor bad. Statistically significant differences were found between the Project condition and all other conditions, in which all replay conditions are rated as good to their performance (coded as 2) and the Say and Project-Say conditions are reported as somewhat good (coded as 1).

In the effort ratings, participants rated projection indicators only (the Project condition) as somewhat high (coded as 1). Statistical significant results are found between Project and each replay condition.

For frustration levels, statistically significant differences were found between each replay and non-replay condition pair. Participants rated replay conditions as low frustration, Say as somewhat low, and Project as well as Project-Say neither low nor high.

In summary, replay conditions and the Say condition generally performed the best.

Thus, **H6.4** is mostly supported. Participants experienced the same workload in the Say and Project conditions for four subscales of mental demand (neutral to somewhat high, *n.s.*), temporal demand (low to somewhat low, *n.s.*), effort (neutral to somewhat high, *n.s.*), and frustration level (somewhat low to neutral). There is only one slight statistical significance ($p < 0.05$) between the Say and Project conditions: the performance subscale, somewhat high in Say vs. neutral in Project.

6.4.6 H6.5. A robot is more trustworthy with projection markers (not supported)

We analyzed Likert responses to the four subscales to measure trust: predictability, reliability, and competence in addition to trust. The ratings are visualized in Figure 60.

Similar to the NASA Taskload questionnaire, we ran Kruskal-Wallis H tests across all subscales and statistically significant results are revealed in all of them: predictability ($\chi^2(6) = 24.8, p < 0.001$), reliability ($\chi^2(6) = 23.9, p < 0.001$), competence ($\chi^2(6) = 49.8, p < 0.0001$), and trust ($\chi^2(6) = 30.7, p < 0.0001$). Then we ran post-hoc Mann-Whitney U pairwise compar-

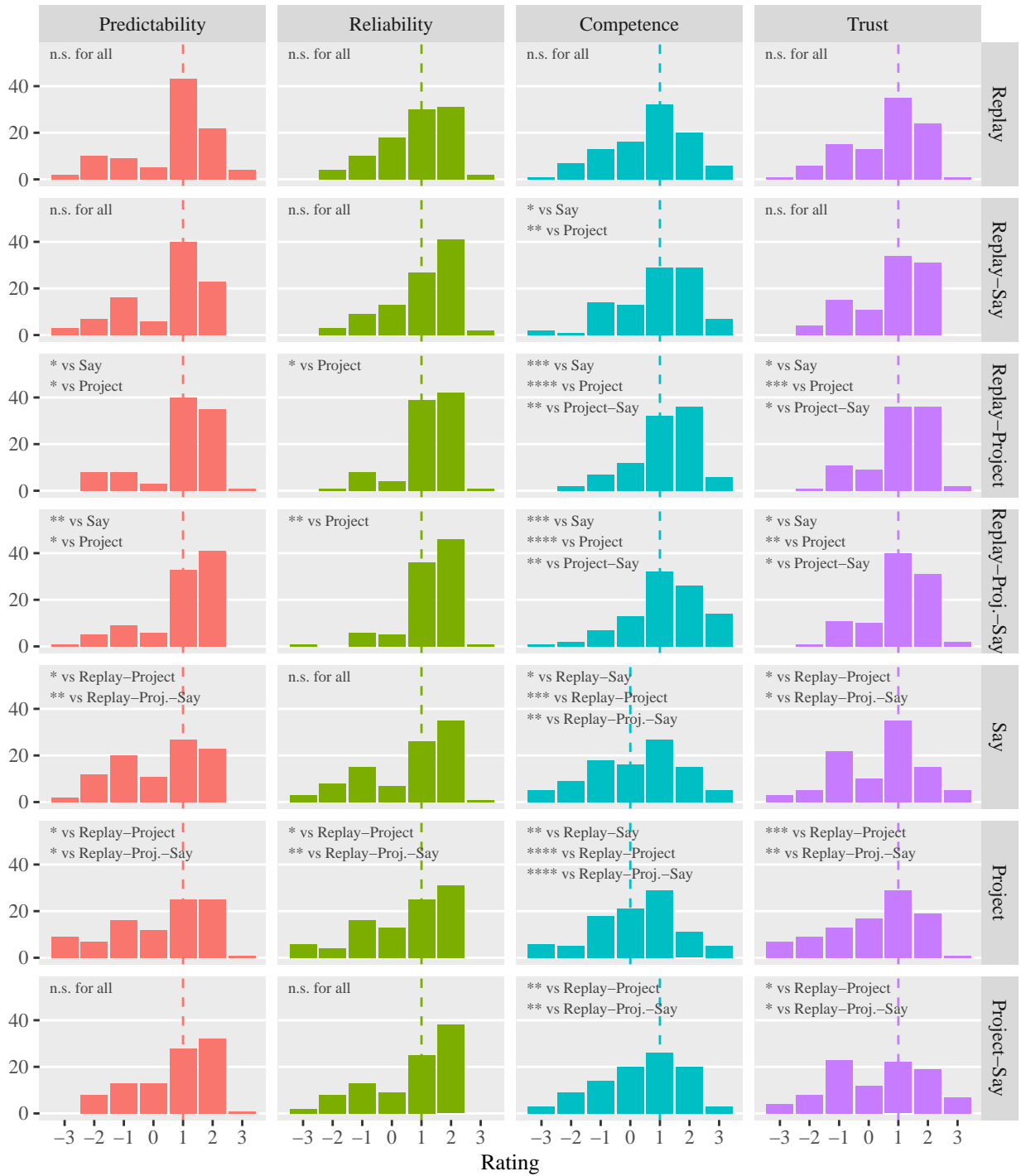


Figure 60: Responses to the Muir trust questionnaire [120]. Regarding predictability, participants rated non-replay conditions less predictable. For reliability, when accompanied with replays or with both replays and verbal indicators, projection markers improve reliability. In terms of competence, adding replay to either Say or Project or both increases the competence rating. For the direct trust measure, replay conditions have more positive ratings than their non-replay counterparts. See Section 6.4.6 for more details.

isons with Holm-Bonferroni correction. The results are also annotated in Figure 60.

In terms of predictability, projection and verbal markers only make the robot more predictable with replays or with both replays and verbal indicators. We found four pairs of statistically slightly significant differences, as seen in the first column of Figure 60, between Replay-Project and Say, between Replay-Project, Replay-Project-Say and Say, Project.

Approximately 25% more participants reported that the verbal indications alone and projection indicators alone (i.e., Say and Project, 50 and 51 participants – 52.6% and 53.7%) are less predictable (somewhat predictable, predictable, and very predictable) than the Replay-Project or the Replay-Project-Say condition (75 and 74 participants – 78.9% and 77.9%) where projection or both projection and verbal indications are included. This suggests that the addition led to the differences. As a result, the number of participants who reported unpredictable decreased half from 34 and 32 (35.8% and 33.7%) in the Replay-Project and the Replay-Project-Say condition conditions to 16 and 15 (16.8% and 15.8%) in the Say and Project conditions.

In the reliability ratings, we reach a similar conclusion as predictability: projection markers, but not verbal markers, only make the robot more reliable with replays or with both replays and verbal indicators. Specifically, two statistically slightly significant differences were found between Replay-Project and Project, and between Replay-Project-Say and Project. Fewer participants reported reliability positively in the Project condition: 83 participants (87.4%) in both Replay-Project and Replay-Project-Say conditions had positive reliability ratings (row 3 and 4 in column 2 of Figure 60) while 22.1% and 28.5% fewer participants, only a total of 62 and 56 participants (65.3% and 58.9%) agreed that they can count the robot to do the job in the Project condition (second-last row in column 2).

For competence (third column in Figure 60), statistically significant differences were found between {Replay-Say, Replay-Project, Replay-Project-Say} and {Say, Project}, which indicate that adding physical replay to either Say or Project or both increases the robot's competence from neutral to somewhat high. In addition, statistical significance also exists in Project-Say versus

Replay-Project or Replay-Project-Say, all of which include projection indicators.

Comparing Replay-Project with Project-Say, in which physical replays *replaced* verbal indicators, 25 (26.3%) more participants positively reported the robot's competence (74 in Replay-Project vs. 49 in Project-Say, 77.9% vs. 51.6%). As seen in Figure 60, this resulted in a more right-skewed distribution for competence in the Replay-Project condition (column 3, row 3) than the one for the Project-Say competence responses (column 3, row 7).

Comparing Replay-Project-Say with Project-Say (column 3, row 4 vs. row 7 in Figure 60), in which physical replay is *added* to projection and verbal indicators, we see almost the same effect (only two participants fewer) that 23 (24%) more participants positively rated the robot's competence (72 in Replay-Project-Say vs. 49 in Project-Say, 75.8% vs. 51.6%).

For the more direct trust ratings, statistically significant differences were found between {Replay-Project, Replay-Project-Say} and non-replay conditions {Say, Project, Project-Say}. More participants reported "trust" (coded as 2) in Replay-Project (36 participants, 37.9%) and Replay-Project-Say (31, 32.6%) than non-replay conditions (15 – 15.8% in Say, 19 – 20% in Project, and 19 – 20% participants in Project-Say).

Thus, **H6.5** is not supported. For the mobile manipulation task in the Projection condition without additional replay or verbal or both indicators, participants rated the robot as less trustworthy, less predictable, less reliable (less extent to predictability), and less competent to do the job.

6.4.7 H6.6. Less workload when presented both verbal and projection markers (almost not supported)

This last hypothesis can be tested the same way as H6.4 with visuals from Figure 59. For the responses in the Project-Say condition, where both verbal and projection markers are presented, only the reported performance is better than the Project condition, where participants reported that their performance is somewhat high versus neutral in the Project conditions.

For other statistically significant differences, Project-Say has higher mental demand (neutral) than the Replay and Replay-Project-Say conditions (Both were rated somewhat low) and has higher frustration level (neutral) than all replay conditions (All were rated low).

In the temporal demand and effort ratings, no significant differences were found.

Thus, **H6.6** is almost not supported. Only in one single pair comparison, which is between Project-Say and Project in the performance metric, Project-Say has a better median rating (somewhat high vs. neutral).

6.5 Discussion and Recommendations

Surprisingly, all of our hypotheses about the use of verbal and projection markers alone were either partially supported (H6.1, H6.2, H6.3 – Efficiency, H6.4, and H6.6) or not supported at all (H6.3 – Accuracy and H6.5), although we believed that instant projection directly onto the operating environment would be more effective and efficient in terms of causal inference as well as mental workload.

In contrast, a combination of physical replays with the verbal and/or projection markers have shown better effectiveness and efficiency as well as lower workload and increased trust. This is consistent with the findings of our previous study [81] that physical arm movement should be accompanied by verbal explanations.

6.6 Effectiveness of inferring missing causal information from the past

With verbal indicators (Section 6.4.1), participants were able to effectively infer where the robot *placed* the gearbox bottom, at the same level as other conditions except for Project, as shown in Figure 50. However, the verbal indicators did not allow participants to infer where the robot *picked* the misrecognized object nor where the ground obstacle was during *navigation*. In both the picking and navigation inferences, half of the participants chose a place near the actual answer, as shown in Figures 48 and 49. Thus, verbal indicators using relative directional words should not be used alone

for picking and navigation inference. Rather, arm movements with the grasping behavior should be used for picking as almost all of the participants in all replay conditions made the correct inference, as shown in Figure 48. Physical replay and projection, instead of speech alone, should be included for better efficiency in navigation tasks because almost all of the participants in all conditions, except for the Say condition, correctly inferred where the ground obstacle was, as shown in Figure 49.

Projection is remarkably good for navigation inferences (Section 6.4.2), on par with replay conditions, but only half of the participants were correct for picking inferences (see the bottom two rows in Figure 48) and no statistical significance was found for placement inferences (see the second last row in Figure 50). Without looking at when the inference happens, projection markers are faster if a person is looking for some causal inference during a robot's navigation task, because physical movements from the robot's base take time to replay. For picking and placing inferences, projection and verbal markers should be present with arm movement to accelerate the causal inference process. Otherwise, these two indicators alone deteriorate people's causal inference performance, shown from the last three rows in Figure 48 and the second-last row in Figure 50.

As shown in Figure 50, placement inference effectiveness is different because almost all participants in at least one condition in the other inference types (picking and navigation) were able to correctly infer the causal information (Figure 48 and 49), but only around 60 (63.2%) in all 95 participants did so for placement inference in every condition except for the Project condition, for which all responses were not significant.

The difficulty with placement inference might be that the gripper did not move into the compartment where the object was dropped. Not moving inside the square caddy compartment, compared with another manipulation subtask – picking, where the robot moved to the target object to be grasped – made it more difficult to infer which caddy section the gripper was above. Even with projection into the caddy section, we might see better results if participants were next to the

robot, in person, to see the projection inside of the caddy, rather than viewing the projection on a recording at a fixed angle. The recommendation here is still to have a multimodal approach with physical replay, verbal and projection indications to inform people into where an object is placed, but we see that additional information might be useful. For example, instead of using an exact replay, an enhanced replay might be better, where the robot would move its gripper into the caddy section into which the object had been dropped, since it is harder for people to make the spatial inference over a set of concave compartments. Future work could investigate cases with placement in concave objects with the robot using additional physical indicators to improve inference-making of where the object was placed.

6.7 Efficiency to infer past missing causal information

As discussed in Section 6.4.3, most participants reported they inferred the *picking* location during the last three events after the robot's arm moved close to the object, as shown in the top four rows in Figure 51. Participants in non-replay conditions, shown in the last three rows of Figure 51, made early inferences and reported before the robot's head started moving, when the head points to the projection or when the robot starts speaking or both. However, 31 (32.6%) of the 95 participants in the Project condition reported they never knew, and there were 11 (11.6%) participants who never knew in each of the other two non-replay conditions, Say and Project-Say, although no statistical significance was found, shown from the top bar in the last three rows in Figure 55. This suggests that both an eye-gazing cue and the verbal/projection indicators are not enough for picking inference; in practice, replay should be included in these cues.

From the analysis of participants' responses to the *navigation* timing questions with three different responses (Figures 52 to 54), two-thirds of the participants were able to make the inference right after seeing the projection. (The other third had no statistical significance; see Figure 54). Verbal indicators alone are the best overall if we only consider statistically significant results (We did see non-significant results from responses, shown from Figure 53).

For the replay conditions, we saw interesting reactions from participants. While the performance in the Replay-Say condition is on par with Replay (top two rows in Figure 52), adding projection indicators delays the inference (the third row in Figure 52). However, with both verbal and projection indicators (Replay-Project-Say) the fourth row in Figure 52), 26 (27.4%) of the 95 participants reported that they knew before the robot started moving, while there were fewer than 10 participants indicating such when only one indicator type was used (as shown by the first bar of the middle two rows in Figure 52).

Therefore, our recommendation for making the navigation inference faster is to only use projection, as shown in Figure 54. This is consistent with previous research on navigation path visualization from other researchers (e.g., [41, 33]). If projection is not available, we recommend using verbal indicators without any replay, shown in Figure 53 rather than simply replaying the base movement.

For placing, we see the same pattern as picking. As shown in Figure 55, participants' inference timing in Replay conditions (top four rows) is distributed to three later events after the robot's gripper is close to the caddy. In the Project condition, there are 31 (32.6%) participants not knowing which caddy section the object was placed into, while 60 (63.2%) participants reported they knew after the robot's head stops moving. The Say and Project-Say condition perform the best: after analyzing responses to the "other" option, ~ 80 participants in both conditions were able to infer after its head stopped moving. So, in practice, to accelerate people's placement inference, verbal indicators should be used.

As the analysis above about effectiveness and efficiency is rather complex, we created Table 8 to ease comprehension and reinforce our recommendations.

6.8 Inference accuracy and confidence of past missing causal information

Regarding the accuracy aspect of inference (Section 6.4.4), the responses show a simpler pattern. In general, replay conditions make participants more confident in their picking and navigation

Table 8: The effectiveness and efficiency of the conditions in inference-making. Shaded boxes are the best values in each column.

| | Picking Inference Where was the object? | | Navigation Inference Where was ground obstacle? | | Placement Inference Which section of caddy? | |
|---------------------------|---|--------------------------------------|---|---|---|--------------------------------------|
| | <i>Effective?</i> ¹ Fig. 48 | <i>When?</i> ² Fig. 51 | <i>Effective?</i> ¹ Fig. 49 | <i>When?</i> ² Fig. 52–54 | <i>Effective?</i> ¹ Fig. 50 | <i>When?</i> ² Fig. 55 |
| Replay | 91.6% | Above table | 83.2% | Towards E | 58.9% | Close to caddy |
| Replay-Say | 97.9% | Above table | 78.9% | Towards E | 60.0% | Over caddy to release |
| Replay-Project | 93.7% | Above table | 92.6% | Towards F (Before E) | 67.4% | Close to caddy |
| Replay-Project-Say | 93.7% | Above table | 91.6% | Towards F & E (Even number) | 56.8% | Over caddy to release |
| Say | 8.4% (49.5% nearby) | Head stopped (22.1% never knew) | 40.0% (49.5% nearby) | Speech | 32.6% | Head stopped & speech |
| Project | 50.5% | Head stopped (42.1% never knew) | 93.7% | N/A ³ (Same as Project-Say below) | 32.6% | Head stopped (32.6% never knew) |
| Project-Say | 55.8% | Head stopped (28.4% never knew) | 83.2% | Projection | 61.1% | Head stopped & speech |

1. See Section 6.4.1 for relevant results and Section 6.6 for a full discussion.

2. The event with highest number of responses is given. See Section 6.4.2 for relevant results and Section 6.7 for a full discussion.

3. There was no timing in the Projection condition since it has projection in the whole video.

inference choices, as shown in Figures 56 and 57. For placement, people are more confident when there are both physical replay and verbal indicators, shown in Figure 58. Thus, we recommend physical replays to increase confidence level for picking and navigation inferences. For placement, we suggest including verbal indicators with physical replay.

6.9 Mental workload for past causal information inference

In terms mental workload (Sections 6.4.5 and 6.4.7), replay conditions generally performed the best. As shown in Figure 60, the Project condition is largely a few Likert item levels worse and the Say condition is also somewhat the same. The reason for the Project case might be that even the projection is directly on the operating environment, its meaning is rather implicit even though the projection itself is explicit. So, in practice, projection indicators should be combined with the physical replay to achieve other benefits discussed throughout this section.

6.10 Perceived trust as a result of causal inference indications

As shown in Figure 60 of Section 6.4.6, conditions with multiple cues have more positive ratings in the trust subscales, including predictability, reliability, competence, and trust. In general, replay conditions perform better than non-replay conditions, and Replay-Project and Replay-Project-Say are among the best two, in which there are always statistically significant results found when compared with non-replay conditions. So, we would recommend the Replay-Project and Replay-Project-Say conditions to achieve slightly better results.

6.11 Limitations and Future Work

In the study, we have placed the camcorder at the best viewing position and angle where a person can view the whole relevant scene and the robot. However, in reality, humans are not static as the camcorder: when not in a crowd, they may move around to better understand the event. We thought of recording multiple videos and placing them in a grid and present them to participants.

However, we had the concern that this may distract their attention or increase workload, requiring them to re-focus as they move from one sub-video to another to look at different angles. Ideally, we can simulate the human walking path by moving the camcorder. However, the walking path would likely be different from person to person because the path is driven by the particular human's thought process, thus it becomes a problem of its own. As future work, online studies with videos from multiple angles would be interesting to pursue, as would in person studies.

In addition, we tried to discretize time by having a fixed set of timing events for picking, navigation, and placing during the mobile manipulation task. In the past [84], we have analyzed the timing by recording videos of participants and extracting the timing information frame by frame. In the future, one may do the same in order to get more accurate data directly from the continuous variable of time.

To measure which method is more accurate, we measured the perceived accuracy using the subjective confidence metric. However, being more confident in inference-making may not be the same as the communication method being accurate. Likewise, these are not necessarily causal of each other. As there will be more research into comparisons of different communications, future researchers can explore different metrics or summarize these comparison work.

Like trust, causal inference-making is a process, rather than a single-time-point event. People may start inferring the past missing causal information at an earlier event but unsure yet, and then become more confident as the robot finishes the task. While we did attempt to capture this by having only part of the task sequence, e.g., only having projection in the Project condition, a better method would be implementing the think-aloud protocol or ask participants to finish the survey during the process to capture the change over time, just like Desai et al. did to study trust [47].

Lastly, as we drew inspiration from the human imitation literature, we can also learn from the teaching literature. It is well-known that learners have preferences in the communication channels, i.e., visual learners and auditory learners [70]. If robots could know this information beforehand, robots can better accommodate individual differences to produce better inference outcomes.

This is particularly important for personal robots. A study into this preference can greatly complement this work where we targeted the general population.

6.12 Conclusion

In this work, we have investigated how a robot could communicate past causal information in a mobile manipulation scenario, encompassing picking, navigation, and placement. Physical replay with head, arm, and base movement is implemented with verbal and projection indicators.

In general, results suggest a multimodal approach: combining physical replay with verbal and projection indicators performs the best in helping participants to infer where the robot picked an object, where the ground obstacle is, and where the robot has placed an object into a caddy. In addition, we found that projection markers alone are remarkably efficient in helping people make navigation inferences, while verbal indicators are exceptionally efficient for making placing inferences.

7 Conclusions and Future Work

7.1 Conclusions and Contributions

This dissertation systematically examined multiple aspects of causal robot explanations. As the dissertation title implies, we treated robot explanation as a process that gradually unfolded.

From the human-subjects study on desired robot explanations, we were able to find many insights into the robot explanations that people prefer, from the need to couple non-verbal cues with verbal explanations, to the preference to address explainees and then provide concise explanations so people can ask a few follow-up questions for more details.

We also contributed explanation generation algorithms, in which roboticists can present robot tasks as action sequences in behavior trees and use our open-sourced algorithms to generate failure or shallow hierarchical explanations, which not only provide succinct explanations but also answer follow-up questions, as informed by the study above.

Finally, we investigated explanation communication with a contribution of implementation of projection mapping, an instant and salient robot-specific communication method that humans do not have. In addition, through another human-subjects experiment, we expanded our knowledge of how a robot could aid people to infer missing causal information in a mobile manipulation task, encompassing picking, navigation, and placement tasks, which resulted in replaced objects. Results suggest a multimodal approach: using all physical replay, verbal, and projection indicators will lead to better aid in inference-making, less mental workload, and more trustworthy robots.

7.2 Future Work

While we scoped the research in this dissertation to focus on important issues in the explanation process, there is still much work to be done to fully achieve causal robot explanations.

In general, the work needs to be expanded to investigate **different types of robots and task domains**. In this dissertation, we have used the manipulator robot Baxter and the mobile manipu-

lator robot Fetch. Mobile robots with or without manipulation capabilities were not investigated. It would be interesting to confirm whether the findings are also applicable to other types of robots. Two major categories are

- Legged robots or humanoids – such as the four-legged Spot from Boston Dynamics and the two-legged humanoid Valkyrie from NASA and Digit from Agility Robotics, and
- Wheeled robots or autonomous vehicles, from small wheeled robots for delivery to a full-scaled Tesla car.

Robot dogs have become very popular – Boston Dynamic’s YouTube videos³⁶ are constant hits on YouTube and have amassed millions of views. However, research with robot dogs is rare in the human-robot interaction literature and they have not been the platform used to study robot explanations. These robot dogs have been typically used without speaking, as animals. However, unlike actual dogs that can be well-understood by their humans, robot dogs might become hard to understand once robot developers program them to exhibit non-dog-like behaviors, because people could not reuse their experience with animal dogs to understand a dog-shaped robot with non-dog-like behaviors.

Mobile robots or autonomous vehicles are increasingly used for delivery and are replacing some manual vehicles. As they autonomously make decisions on the sidewalks and roads, which were mainly built for pedestrians and human drivers, these autonomous entities need to estimate the understandability of their resulted behaviors from their decisions. Unlike a worker who might be trained how to work with a robot, pedestrians and drivers are bystanders who may have limited knowledge of the robots.

With the current push towards legged robots, mobile robots, and autonomous vehicles, we need to expand our knowledge of what people understand or do not understand about these robots

³⁶<https://www.youtube.com/channel/UC7vVhKEfw4nOGp8TyDk7RcQ>

and how could the robots could provide explanations to improve people's understanding of the systems.

In addition to investigating different types of robots, we also need to investigate different task domains and applications. To make our findings more generalizable, we have investigated both manipulation and navigation tasks in a common handover and an assembly scenario. However, as robots become ubiquitous, there are different task domains and interaction scenarios that may present new challenges for robot explanations. Examples include social robots providing support in retail stores, service robots delivering dishes in restaurants, and underwater robots used for inspection.

Additional future research should be conducted on **proactive explanations**. In this dissertation, robots act as passive agents: they offer explanations with the assumption that humans will understand or they will ask follow-up questions, i.e., human-initiated explanations. What is not known yet is, as I just mentioned, how robots could estimate the understandability of the resulted behaviors from their decisions and offer explanations proactively. This not only requires their understandability estimate to initiate explanation but also estimating humans' understanding of those explanations to proactively continue explaining, rather than waiting for the explainee to ask follow-up questions.

While this may sound like long-term work, a good starting point might be conducting a human-subjects study to experiment with a set of first explanations and, after a short break, follow up explanations, which simulates the scenario where a human is trying to understand the first explanation. This experiment could also offer different follow-up explanations to estimate the understandability of the first explanation. With the results and data analysis, we could then find a way to encode such explanation understandability estimates in algorithms.

We hope that this dissertation has laid the foundation and paves the way to improved human-robot interaction, powered by robots capable of providing explanations about their actions and capabilities.

Literature Cited

- [1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 582:1–582:18.
- [2] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 582:1–582:18.
- [3] David Abel. 2019. simple_rl: Reproducible Reinforcement Learning in Python. In *ICLR Workshop on Reproducibility in Machine Learning*. Available at https://github.com/david-abel/simple_rl.
- [4] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.
- [5] Henny Admoni and Brian Scassellati. 2015. Robot Nonverbal Communication as an AI Problem (and Solution). In *2015 AAAI Fall Symposium Series*.
- [6] Henny Admoni and Brian Scassellati. 2017. Social eye gaze in human-robot interaction: a review. *Journal of Human-Robot Interaction* 6, 1 (2017), 25–63.
- [7] Henny Admoni, Thomas Weng, Bradley Hayes, and Brian Scassellati. 2016. Robot non-verbal behavior improves task performance in difficult collaborations. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. 51–58.
- [8] Dan Amir and Ofra Amir. 2018. Highlights: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2018)*. 1168–1176.
- [9] Ofra Amir, Finale Doshi-Velez, and David Sarne. 2018. Agent Strategy Summarization. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems (AAAMS)*. International Foundation for Autonomous Agents and Multiagent Systems, 1203–1207.
- [10] Rasmus S Andersen, Ole Madsen, Thomas B Moeslund, and Heni Ben Amor. 2016. Projecting robot intentions into human environments. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 294–301.
- [11] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. 2019. Explainable agents and robots: Results from a systematic literature review. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1078–1088.

- [12] Alexander Mois Aroyo, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. 2018. Trust and social engineering in human robot interaction: Will a robot make you disclose sensitive information, conform to its recommendations or gamble? *IEEE Robotics and Automation Letters* 3, 4 (2018), 3701–3708.
- [13] J Andrew Bagnell, Felipe Cavalcanti, Lei Cui, Thomas Galluzzo, Martial Hebert, Moslem Kazemi, Matthew Klingensmith, Jacqueline Libby, Tian Yu Liu, Nancy Pollard, et al. 2012. An integrated system for autonomous robotics manipulation. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2955–2962.
- [14] W. A. Bainbridge, J. Hart, E. S. Kim, and B. Scassellati. 2008. The effect of presence on human-robot interaction. In *The 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 701–706.
- [15] F. Balint-Benczédi, Z. Márton, M. Durner, and M. Beetz. 2017. Storing and retrieving perceptual episodic memories for long-term manipulation tasks. In *2017 18th International Conference on Advanced Robotics (ICAR)*. 25–31.
- [16] Albert Bandura. 1999. Social cognitive theory: An agentic perspective. *Asian journal of social psychology* 2, 1 (1999), 21–41.
- [17] Albert Bandura. 2008. *Observational Learning*. American Cancer Society.
- [18] Rachel Barr and Nancy Wyss. 2008. Reenactment of televised content by 2-year olds: Toddlers use language learned from television to solve a difficult imitation problem. *Infant Behavior and Development* 31, 4 (2008), 696–703.
- [19] Michael Beetz, Daniel Beßler, Andrei Haidu, Mihai Pomarlan, Asil Kaan Bozcuoğlu, and Georg Bartels. 2018. Know Rob 2.0 – A 2nd Generation Knowledge Processing Framework for Cognition-Enabled Robotic Agents. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 512–519. Available at <https://github.com/knowrob/knowrob/>.
- [20] Michael Beetz, Lorenz Mösenlechner, and Moritz Tenorth. 2010. CRAM — A Cognitive Robot Abstract Machine for everyday manipulation in human environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1012–1017.
- [21] Michael Beetz, Moritz Tenorth, and Jan Winkler. 2015. Open-EASE — A Knowledge Processing Service for Robots and Robotics/AI Researchers. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 1983–1990.
- [22] Gisela Böhm and Hans-Rüdiger Pfister. 2015. How people explain their own and others’ behavior: a theory of lay causal explanations. *Frontiers in Psychology* 6 (2015), 139.
- [23] Jonathan Bohren and Steve Cousins. 2010. The SMACH high-level executive. *IEEE Robotics & Automation Magazine* 17, 4 (2010), 18–20.

- [24] Jonathan Bohren, Radu Bogdan Rusu, E Gil Jones, Eitan Marder-Eppstein, Caroline Pantofaru, Melonee Wise, Lorenz Mösenlechner, Wim Meeussen, and Stefan Holzer. 2011. Towards autonomous robotic butlers: Lessons learned with the PR2. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5568–5575.
- [25] Daniel J Brooks. 2017. *A Human-Centric Approach to Autonomous Robot Failures*. Ph.D. Dissertation. Ph. D. dissertation, Department of Computer Science, University.
- [26] Daniel J Brooks, Momotaz Begum, and Holly A Yanco. 2016. Analysis of reactions towards failures and recovery strategies for autonomous robots. In *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 487–492.
- [27] Sebastian G Brunner, Peter Lehner, Martin J Schuster, Sebastian Riedel, Rico Belder, Daniel Leidner, Armin Wedler, Michael Beetz, and Freerk Stulp. 2018. Design, execution, and post-mortem analysis of prolonged autonomous robot operations. *IEEE Robotics and Automation Letters* 3, 2 (2018), 1056–1063.
- [28] Sebastian G Brunner, Franz Steinmetz, Rico Belder, and Andreas Dömel. 2016. RAFCON: A graphical tool for engineering complex, robotic tasks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3283–3290.
- [29] Barbara Bruno, Roberto Menicatti, Carmine T Recchiuto, Edouard Lagrue, Amit K Pandey, and Antonio Sgorbissa. 2018. Culturally-Competent Human-Robot Verbal Interaction. In *2018 15th International Conference on Ubiquitous Robots (UR)*. 388–395.
- [30] Daphna Buchsbaum, Alison Gopnik, Thomas L Griffiths, and Patrick Shafto. 2011. Children’s imitation of causal action sequences is influenced by statistical and pedagogical evidence. *Cognition* 120, 3 (2011), 331–340.
- [31] David Buttelmann, Andy Schieler, Nicole Wetzels, and Andreas Widmann. 2017. Infants’ and adults’ looking behavior does not indicate perceptual distraction for constrained modelled actions- An eye-tracking study. *Infant behavior and development* 47 (2017), 103–111.
- [32] Elizabeth Cha, Yunkyung Kim, Terrence Fong, Maja J Mataric, et al. 2018. A survey of nonverbal signaling methods for non-humanoid robots. *Foundations and Trends® in Robotics* 6, 4 (2018), 211–323. PDF is available at https://www.lizcha.com/publications/ft_2018.pdf.
- [33] Ravi Teja Chadalavada, Henrik Andreasson, Robert Krug, and Achim J Lilienthal. 2015. That’s on my mind! robot to human intention communication through on-board projection on shared floor space. In *2015 European Conference on Mobile Robots (ECMR)*. 1–6.
- [34] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. 2020. The emerging landscape of explainable automated planning & decision making. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. 4803–4811.

- [35] Tathagata Chakraborti, Sarath Sreedharan, Anagha Kulkarni, and Subbarao Kambhampati. 2018. Projection-aware task planning and execution for human-in-the-loop operation of robots in a mixed-reality workspace. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4476–4482.
- [36] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*. 156–163.
- [37] Angelos Chatzimpampas, Rafael M Martins, Ilir Jusufi, and Andreas Kerren. 2020. A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization* 19, 3 (2020), 207–233.
- [38] Sachin Chitta, Ioan Sucan, and Steve Cousins. 2012. MoveIt! *IEEE Robotics & Automation Magazine* 19, 1 (2012), 18–19. Software available at <https://moveit.ros.org>.
- [39] Michele Colledanchise and Petter Ögren. 2016. How behavior trees modularize hybrid control systems and generalize sequential behavior compositions, the subsumption architecture, and decision trees. *IEEE Transactions on Robotics* 33, 2 (2016), 372–389.
- [40] Michele Colledanchise and Petter Ögren. 2018. *Behavior Trees in Robotics and AI: An Introduction* (1st ed.). CRC Press, Boca Raton, FL, USA.
- [41] Michael D Covert, Tiffany Lee, Ivan Shindeev, and Yu Sun. 2014. Spatial augmented reality as a method for a mobile robot to communicate intended movement. *Computers in Human Behavior* 34 (2014), 241–248.
- [42] Mike Daily, Youngkwan Cho, Kevin Martin, and Dave Payton. 2003. World embedded interfaces for human-robot interaction. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*. IEEE, 6–pp.
- [43] Devleena Das, Siddhartha Banerjee, and Sonia Chernova. 2021. Explainable ai for robot failures: Generating explanations that improve user assistance in fault recovery. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. 351–360.
- [44] Maartje MA de Graaf and Bertram F Malle. 2017. How people explain action (and autonomous intelligent systems should too). In *2017 AAAI Fall Symposium Series*.
- [45] Michiel de Jong, Kevin Zhang, Aaron M Roth, Travers Rhodes, Robin Schmucker, Chenghui Zhou, Sofia Ferreira, João Cartucho, and Manuela Veloso. 2018. Towards a robust interactive and learning social robot. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 883–891.

- [46] Munjal Desai, Poornima Kaniarasu, Mikhail Medvedev, Aaron Steinfeld, and Holly Yanco. 2013. Impact of Robot Failures and Feedback on Real-time Trust. In *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction*. 251–258.
- [47] Munjal Desai, Poornima Kaniarasu, Mikhail Medvedev, Aaron Steinfeld, and Holly Yanco. 2013. Impact of robot failures and feedback on real-time trust. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 251–258.
- [48] Munjal Desai, Mikhail Medvedev, Marynel Vázquez, Sean McSheehy, Sofia Gadea-Omelchenko, Christian Bruggeman, Aaron Steinfeld, and Holly Yanco. 2012. Effects of changing reliability on trust of robot systems. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 73–80.
- [49] Robert F DeVellis. 2016. *Scale development: Theory and applications*. Vol. 26. SAGE.
- [50] Sandra Devin and Rachid Alami. 2016. An implemented theory of mind to improve human-robot shared plans execution. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 319–326.
- [51] André Dietrich, Sebastian Zug, and Jörg Kaiser. 2015. Selectscript: A query language for robotic world models and simulations. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 6254–6260.
- [52] André Dietrich, Sebastian Zug, Siba Mohammad, and Jörg Kaiser. 2014. Distributed management and representation of data and context in robotic applications. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1133–1140.
- [53] Thomas G Dietterich. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* 13 (2000), 227–303.
- [54] Anca Dragan and Siddhartha Srinivasa. 2014. Familiarization to robot motion. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 366–373.
- [55] Anca D Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S Srinivasa. 2015. Effects of robot motion on human-robot collaboration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 51–58.
- [56] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. 2013. Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. 301–308.
- [57] Maximilian Durner, Simon Kriegel, Sebastian Riedel, Manuel Brucker, Zoltán-Csaba Márton, Ferenc Bálint-Benczédi, and Rudolph Triebel. 2017. Experience-based optimization of robotic perception. In *2017 18th International Conference on Advanced Robotics (ICAR)*. 32–39.

- [58] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O Riedl. 2019. Automated rationale generation: a technique for explainable AI and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI)*. 263–274.
- [59] Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner. 2007. G* Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods* 39, 2 (2007), 175–191. Software available at <http://www.gpower.hhu.de/>.
- [60] Ronit Feingold-Polak, Avital Elishay, Yonat Shahar, Maayan Stein, Yael Edan, and Shelly Levy-Tzedek. 2018. Differences between young and old users when interacting with a humanoid robot: a qualitative usability study. *Paladyn, Journal of Behavioral Robotics* 9, 1 (2018), 183–192.
- [61] Naomi T Fitter and Katherine J Kuchenbecker. 2016. Designing and assessing expressive open-source faces for the baxter robot. In *International Conference on Social Robotics*. 340–350.
- [62] Cliff Fitzgerald. 2013. Developing baxter. In *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 1–6.
- [63] Tully Foote. 2013. tf: The transform library. In *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 1–6.
- [64] Dehann Fourie, Samuel Claassens, Sudeep Pillai, Roxana Mata, and John Leonard. 2017. SLAMinDB: Centralized graph databases for mobile robotics. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 6331–6337.
- [65] Kevin French, Shiyu Wu, Tianyang Pan, Zheming Zhou, and Odest Chadwicke Jenkins. 2019. Learning Behavior Trees From Demonstration. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 7791–7797.
- [66] Cheryl D Fryar, Deanna Kruszan-Moran, Qiuping Gu, and Cynthia L Ogden. 2018. Mean body weight, weight, waist circumference, and body mass index among adults: United States, 1999–2000 through 2015–2016. *National Health Statistics Reports* 122 (2018), 1–16.
- [67] Yuxiang Gao and Chien-Ming Huang. 2019. PATI: a projection-based augmented table-top interface for robot programming. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 345–355.
- [68] Amy K Gardiner, Marissa L Greif, and David F Bjorklund. 2011. Guided by intention: Preschoolers’ imitation reflects inferences of causation. *Journal of Cognition and Development* 12, 3 (2011), 355–373.

- [69] Fabrizio Ghiringhelli, Jérôme Guzzi, Gianni A Di Caro, Vincenzo Caglioti, Luca M Gambardella, and Alessandro Giusti. 2014. Interactive augmented reality for understanding and analyzing multi-robot systems. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1195–1201.
- [70] Abbas Pourhossein Gilakjani et al. 2012. Visual, auditory, kinaesthetic learning styles and their impacts on English language teaching. *Journal of studies in education* 2, 1 (2012), 104–113.
- [71] Alison Gopnik. 2000. Explanation as orgasm and the drive for causal knowledge: The function, evolution, and phenomenology of the theory formation system. In *Explanation and cognition*, Frank C. Keil and Robert Andrew Wilson (Eds.). The MIT Press, Chapter 12, 299–323.
- [72] Alison Gopnik, Laura Schulz, and Laura Elizabeth Schulz. 2007. *Causal learning: Psychology, philosophy, and computation*. Oxford University Press.
- [73] Michael Görner, Robert Haschke, Helge Ritter, and Jianwei Zhang. 2019. MoveIt! task constructor for task-level motion planning. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 190–196.
- [74] David Gossow, Adam Leeper, Dave Hershberger, and Matei Ciocarlie. 2011. Interactive Markers: 3-D User Interfaces for ROS Applications. *IEEE Robotics & Automation Magazine* 18, 4 (2011), 14–15.
- [75] Anselm Grundhöfer and Daisuke Iwai. 2018. Recent advances in projection mapping algorithms, hardware and applications. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 653–675.
- [76] Kelleher R Guerin, Colin Lea, Chris Paxton, and Gregory D Hager. 2015. A framework for end-user instruction of a robot assistant for manufacturing. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6167–6174.
- [77] Pascal Haazebroek, Saskia van Dantzig, and Bernhard Hommel. 2011. A computational model of perception and action for cognitive robotics. *Cognitive processing* 12, 4 (2011), 355.
- [78] H Haidarian, W Dinalankara, S Fults, Shomir Wilson, Don Perlis, M Schmill, T Oates, D Josyula, and M Anderson. 2010. The metacognitive loop: An architecture for building robust intelligent systems. In *PAAAI Fall Symposium on Commonsense Knowledge (AAAI/CSK'10)*.
- [79] Zhao Han, Jordan Allspaw, Gregory LeMasurier, Jenna Parrillo, Daniel Giger, S Reza Ahmadzadeh, and Holly A Yanco. 2020. Towards Mobile Multi-Task Manipulation in a Confined and Integrated Environment with Irregular Objects. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 11025–11031.

- [80] Zhao Han, Jordan Allspaw, Adam Norton, and Holly A Yanco. 2019. Towards A Robot Explanation System: A Survey and Our Approach to State Summarization, Storage and Querying, and Human Interface. In *Proceedings of The Artificial Intelligence for Human-Robot Interaction (AI-HRI) Symposium at AAI Fall Symposium Series (AAAI-FSS) 2019*. arXiv:1909.06418
- [81] ZHAO HAN, PHILLIPS ELIZABETH, and HOLLY A YANCO. 2021. The Need for Verbal Robot Explanations and How People Would Like a Robot To Explain Itself. *ACM Transactions on Human-Robot Interaction* (2021). Accepted, awaiting publication.
- [82] Zhao Han, Daniel Giger, Jordan Allspaw, Michael S Lee, Henny Admoni, and Holly A Yanco. 2021. Building The Foundation of Robot Explanation Generation Using Behavior Trees. *ACM Transactions on Human-Robot Interaction* (2021). Accepted, awaiting publication.
- [83] Zhao Han, Alexander Wilkinson, Jenna Parrillo, Jordan Allspaw, and Holly A Yanco. 2020. Projection Mapping Implementation: Enabling Direct Externalization of Perception Results and Action Intent to Improve Robot Explainability. *The Artificial Intelligence for Human-Robot Interaction Symposium at AAI Fall Symposium Series 2020 (AI-HRI)* (2020).
- [84] Zhao Han and Holly Yanco. 2019. The Effects of Proactive Release Behaviors During Human-Robot Handovers. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 440–448.
- [85] Zhao Han and Holly A Yanco. 2020. Reasons People Want Explanations After Unrecoverable Pre-Handover Failures. *ICRA 2020 Workshop on Human-Robot Handovers* (2020).
- [86] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.
- [87] Bradley Hayes and Julie A Shah. 2017. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*. 303–312.
- [88] Stefanie Hoehl, Stefanie Keupp, Hanna Schleihauf, Nicola McGuigan, David Buttelmann, and Andrew Whiten. 2019. ‘Over-imitation’: A review and appraisal of a decade of research. *Developmental Review* 51 (2019), 90–108.
- [89] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6, 2 (1979), 65–70.
- [90] Keith James Holyoak and Robert G Morrison. 2012. *The Oxford handbook of thinking and reasoning*. Oxford University Press.

- [91] Shanee Honig and Tal Oron-Gilad. 2018. Understanding and resolving failures in human-robot interaction: Literature review and model development. *Frontiers in psychology* 9 (2018), 861.
- [92] Larry J Hornbeck. 1997. Digital light processing for high-brightness high-resolution applications. In *Projection Displays III*, Vol. 3013. International Society for Optics and Photonics, 27–40.
- [93] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. 2013. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots* (2013). <https://doi.org/10.1007/s10514-012-9321-0> Software available at <http://octomap.github.com>.
- [94] Se-Yeon Jeong, I-Ju Choi, Yeong-Jin Kim, Yong-Min Shin, Jeong-Hun Han, Goo-Hong Jung, and Kyoung-Gon Kim. 2017. A study on ros vulnerabilities and countermeasure. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. 147–148.
- [95] Alisa Kalegina, Grace Schroeder, Aidan Allchin, Keara Berlin, and Maya Cakmak. 2018. Characterizing the design space of rendered robot faces. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 96–104.
- [96] Frank Kaptein, Joost Broekens, Koen Hindriks, and Mark Neerincx. 2017. Personalised self-explanation by robots: The role of goals versus beliefs in robot-action explanation for children and adults. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 676–682.
- [97] Ben Kenward. 2012. Over-imitating preschoolers believe unnecessary actions are normative and enforce their performance by a third party. *Journal of experimental child psychology* 112, 2 (2012), 195–207.
- [98] Annette M Klein, Petra Hauf, and Gisa Aschersleben. 2006. The role of action effects in 12-month-olds’ action control: A comparison of televised model and live model. *Infant Behavior and Development* 29, 4 (2006), 535–544.
- [99] Barbara Koslowski. 1996. *Theory and evidence: The development of scientific reasoning*. MIT Press.
- [100] Minae Kwon, Sandy H Huang, and Anca D Dragan. 2018. Expressing Robot Incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 87–95.
- [101] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.
- [102] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. 2017. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics* 5, 4 (2017), 261–349.

- [103] Chong-U Lim, Robin Baumgarten, and Simon Colton. 2010. Evolving behaviour trees for the commercial game DEFCON. In *European Conference on the Applications of Evolutionary Computation*. Springer, 100–110.
- [104] Ask Media Group LLC. 2019. *How Long Is the Average Human Arm?* <https://www.reference.com/science/long-average-human-arm-62c7536c5e56f385> Accessed on 2019-08-15.
- [105] Tania Lombrozo. 2006. The structure and function of explanations. *Trends in Cognitive Sciences* 10, 10 (2006), 464–470.
- [106] Derek E Lyons, Andrew G Young, and Frank C Keil. 2007. The hidden structure of overimitation. *Proceedings of the National Academy of Sciences* 104, 50 (2007), 19751–19756.
- [107] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. 2020. The Marathon 2: A Navigation System. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2718–2725.
- [108] Bertram F Malle. 2006. *How the mind explains behavior: Folk explanations, meaning, and social interaction*. MIT Press.
- [109] Eitan Marder-Eppstein, Eric Berger, Tully Foote, Brian Gerkey, and Kurt Konolige. 2010. The office marathon: Robust navigation in an indoor office environment. In *2010 IEEE International Conference on Robotics and Automation*. IEEE, 300–307.
- [110] Alejandro Marzinotto, Michele Colledanchise, Christian Smith, and Petter Ögren. 2014. Towards a unified behavior trees framework for robot control. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5420–5427.
- [111] Akihiro Matsufuji and Angelica Lim. 2021. Perceptual Effects of Ambient Sound on an Artificial Agent’s Rate of Speech. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. 67–70.
- [112] Nikolaos Mavridis. 2015. A review of verbal and non-verbal human–robot interactive communication. *Robotics and Autonomous Systems* 63 (2015), 22–35.
- [113] Tim Miller. 2018. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* (2018).
- [114] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38.
- [115] Grégoire Milliez, Raphaël Lallement, Michelangelo Fiore, and Rachid Alami. 2016. Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring. In *2016 11th ACM/IEEE International Conference on Human Robot Interaction (HRI)*. IEEE, 43–50.

- [116] Brent Mittelstadt, Chris Russell, and Sandra Wachter. 2019. Explaining explanations in AI. In *Proceedings of the conference on fairness, accountability, and transparency*. 279–288.
- [117] Daniel E Moerman. 2002. *Meaning, Medicine, and the “Placebo Effect”*. Vol. 28. Cambridge University Press Cambridge.
- [118] AJung Moon, Daniel M Troniak, Brian Gleeson, Matthew KXJ Pan, Minhua Zheng, Benjamin A Blumer, Karon MacLean, and Elizabeth A Croft. 2014. Meet me where i’m gazing: how shared attention gaze affects human-robot handover timing. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. 334–341.
- [119] Daniel Moreno and Gabriel Taubin. 2012. Simple, accurate, and robust projector-camera calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 464–471.
- [120] Bonnie M Muir. 1987. Trust between humans and machines, and the design of decision aids. *International journal of man-machine studies* 27, 5-6 (1987), 527–539.
- [121] Donna L Mumme and Anne Fernald. 2003. The infant as onlooker: Learning from emotional reactions observed in a television scenario. *Child development* 74, 1 (2003), 221–237.
- [122] Hirenkumar Nakawala, Paulo JS Goncalves, Paolo Fiorini, Giancarlo Ferrigno, and Elena De Momi. 2018. Approaches for action sequence representation in robotics: a review. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5666–5671.
- [123] Allen Newell et al. 1982. The knowledge level. *Artificial intelligence* 18, 1 (1982), 87–127.
- [124] Hai Nguyen, Matei Ciocarlie, Kaijen Hsiao, and Charles C Kemp. 2013. Ros commander (rosco): Behavior creation for home robots. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 467–474.
- [125] Raymond S Nickerson. 1999. How we know—and sometimes misjudge—what others know: Imputing one’s own knowledge to others. *Psychological bulletin* 125, 6 (1999), 737.
- [126] Tim Niemueller, Nichola Abdo, Andreas Hertle, Gerhard Lakemeyer, Wolfram Burgard, and Bernhard Nebel. 2013. Towards deliberative active perception using persistent memory. In *Proc. IROS 2013 Workshop on AI-based Robotics*.
- [127] Tim Niemueller, Gerhard Lakemeyer, and Siddhartha S Srinivasa. 2012. A generic robot database and its application in fault analysis and performance evaluation. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 364–369.
- [128] Nils J Nilsson. 1973. *A hierarchical robot planning and execution system*. SRI International.

- [129] Miguel Oliveira, Gi Hyun Lim, Luis Seabra Lopes, S Hamidreza Kasaei, Ana Maria Tomé, and Aneesh Chauhan. 2014. A perceptual memory system for grounding semantic representations in intelligent service robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2216–2223.
- [130] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
- [131] P. F. Palamara, V. A. Ziparo, L. Iocchi, D. Nardi, P. Lima, and H. Costelha. 2008. A Robotic Soccer Passing Task Using Petri Net Plans. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Demo Papers (AAMAS '08)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1711–1712.
- [132] Stefan Palan and Christian Schitter. 2018. Prolific.ac—A subject pool for online experiments. *Journal of Behavioral and Experimental Finance* 17 (2018), 22–27. Available at <https://prolific.co/>.
- [133] Chris Paxton, Andrew Hundt, Felix Jonathan, Kelleher Guerin, and Gregory D Hager. 2017. CoSTAR: Instructing collaborative robots with behavior trees and vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 564–571.
- [134] Chris Paxton, Felix Jonathan, Andrew Hundt, Bilge Mutlu, and Gregory D Hager. 2018. Evaluating methods for end-user creation of robot task plans. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 6086–6092.
- [135] Chris Paxton, Nathan Ratliff, Clemens Eppner, and Dieter Fox. 2019. Representing Robot Task Plans as Robust Logical-Dynamical Systems. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. To appear.
- [136] Elizabeth Phillips, Xuan Zhao, Daniel Ullman, and Bertram F Malle. 2018. What is Human-like? Decomposing Robots’ Human-like Appearance Using the Anthropomorphic roBOT (ABOT) Database. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 105–113.
- [137] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*. 5.
- [138] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, Vol. 3. Kobe, Japan, 5.
- [139] Rubanraj Ravichandran, Erwin Prassler, Nico Huebel, and Sebastian Blumenthal. 2018. A Workbench for Quantitative Comparison of Databases in Multi-Robot Applications. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3744–3750.

- [140] Michael A. Rosen, Eduardo Salas, Davin Pavlas, Randy Jensen, Dan Fu, and Donald Lampton. 2010. Demonstration-Based Training: A Review of Instructional Features. *Human Factors* 52, 5 (2010), 596–609.
- [141] Ismael Sagredo-Olivenza, Pedro Pablo Gómez-Martín, Marco Antonio Gómez-Martín, and Pedro Antonio González-Calero. 2017. Trained behavior trees: Programming by demonstration to support ai game designers. *IEEE Transactions on Games* 11, 1 (2017), 5–14.
- [142] Fred B Schneider. 1990. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys (CSUR)* 22, 4 (1990), 299–319.
- [143] Jan Maarten Schraagen, Susan F Chipman, and Valerie L Shalin. 2000. *Cognitive task analysis*. Psychology Press.
- [144] Trenton Schulz, Jim Torresen, and Jo Herstad. 2019. Animation Techniques in Human-Robot Interaction User Studies: A Systematic Literature Review. *ACM Transactions on Human-Robot Interaction (THRI)* 8, 2 (2019), 12.
- [145] Martin J Schuster, Sebastian G Brunner, Kristin Bussmann, Stefan Büttner, Andreas Dömel, Matthias Hellerer, Hannah Lehner, Peter Lehner, Oliver Porges, Josef Reill, et al. 2019. Towards Autonomous Planetary Exploration: The Lightweight Rover Unit (LRU), its Success in the SpaceBotCamp Challenge, and Beyond. *Journal of Intelligent & Robotic Systems* 93, 3-4 (2019), 461–494.
- [146] Stela H. Seo, Denise Geiskkovitch, Masayuki Nakane, Corey King, and James E. Young. 2015. Poor Thing! Would You Feel Sorry for a Simulated Robot?: A Comparison of Empathy Toward a Physical and a Simulated Robot. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. 125–132.
- [147] David Silvera-Tawil, David Rye, and Mari Velonaki. 2015. Artificial skin and tactile sensing for socially interactive robots: A review. *Robotics and Autonomous Systems* 63 (2015), 230–243.
- [148] Kristyn Sommer, Rebecca Davidson, Kristy L Armitage, Virginia Slaughter, Janet Wiles, and Mark Nielsen. 2020. Preschool children overimitate robots, but do so less than they overimitate humans. *Journal of Experimental Child Psychology* 191 (2020), 104702.
- [149] Sonja Stange and Stefan Kopp. 2020. Effects of a Social Robot’s Self-Explanations on How Humans Understand and Evaluate Its Behavior. In *Proceedings of the 2020 ACM/IEEE international conference on human-robot interaction*. 619–627.
- [150] Aaron Steinfeld. 1999. *The benefit to the deaf of real-time captions in a mainstream classroom environment*. University of Michigan.
- [151] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT Press.

- [152] Daniel Szafrir, Bilge Mutlu, and Terrence Fong. 2015. Communicating directionality in flying robots. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 19–26.
- [153] Aaquib Tabrez, Matthew B Luebbbers, and Bradley Hayes. 2020. A Survey of Mental Modeling Techniques in Human–Robot Teaming. *Current Robotics Reports* (2020), 1–9.
- [154] Paul J Taylor, Darlene F Russ-Eft, and Daniel WL Chan. 2005. A meta-analytic review of behavior modeling training. *Journal of applied psychology* 90, 4 (2005), 692.
- [155] Moritz Tenorth and Michael Beetz. 2009. KnowRob – knowledge processing for autonomous personal robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4261–4266. Available at <https://github.com/knowrob/knowrob/>.
- [156] Moritz Tenorth, Jan Winkler, Daniel Beßler, and Michael Beetz. 2015. Open-EASE: A cloud-based knowledge service for autonomous learning. *KI-Künstliche Intelligenz* 29, 4 (2015), 407–411.
- [157] Sam Thellman, Annika Silvervarg, and Tom Ziemke. 2017. Folk-psychological interpretation of human vs. humanoid robot behavior: exploring the intentional stance toward robots. *Frontiers in Psychology* 8 (2017), 1962.
- [158] W3C. 2009. *OWL 2 Web Ontology Language Document Overview*. W3C Recommendation. W3C. <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/>.
- [159] Sebastian Wallkötter, Silvia Tulli, Ginevra Castellano, Ana Paiva, and Mohamed Chetouani. 2020. Explainable agents through social cues: A review. *arXiv preprint arXiv:2003.05251* (2020).
- [160] Dian Wang, Colin Kohler, Andreas ten Pas, Alexander Wilkinson, Maozhi Liu, Holly Yanco, and Robert Platt. 2018. Towards Assistive Robotic Pick and Place in Open World Environments. *arXiv preprint arXiv:1809.09541* (2018).
- [161] Lujia Wang, Ming Liu, Max Q-H Meng, and Roland Siegwart. 2012. Towards real-time multi-sensor information retrieval in cloud robotic system. In *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. 21–26.
- [162] Ning Wang, David V Pynadath, and Susan G Hill. 2016. The impact of pomdp-generated explanations on trust and performance in human-robot teams. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 997–1005.
- [163] Ning Wang, David V Pynadath, and Susan G Hill. 2016. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 109–116.

- [164] Auriel Washburn, Akanimoh Adeleye, Thomas An, and Laurel D Riek. 2020. Robot errors in proximate HRI: how functionality framing affects perceived reliability and trust. *ACM Transactions on Human-Robot Interaction (THRI)* 9, 3 (2020), 1–21.
- [165] Atsushi Watanabe, Tetsushi Ikeda, Yoichi Morales, Kazuhiko Shinozawa, Takahiro Miyashita, and Norihiro Hagita. 2015. Communicating robotic navigational intentions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 5763–5769.
- [166] Henry M. Wellman, Anne K. Hickling, and Carolyn A. Schult. 1997. Young Children’s Psychological, Physical, and Biological Explanations. *New Directions for Child and Adolescent Development* 1997, 75 (1997), 7–26.
- [167] Andrew Whiten, Gillian Allan, Siobahn Devlin, Natalie Kseib, Nicola Raw, and Nicola McGuigan. 2016. Social learning in the real-world: ‘Over-imitation’ occurs in both children and adults unaware of participation in an experiment and independently of social interaction. *PloS one* 11, 7 (2016), e0159920.
- [168] Christopher D Wickens, Justin G Hollands, Simon Banbury, and Raja Parasuraman. 2015. *Engineering psychology and human performance*. Psychology Press.
- [169] Jan Winkler, Moritz Tenorth, Asil Kaan Bozcuoglu, and Michael Beetz. 2014. CRAMm—memories for robots performing everyday manipulation activities. *Advances in Cognitive Systems* 3 (2014), 47–66.
- [170] Melonee Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich. 2016. Fetch & Freight: Standard Platforms for Service Robot Applications. In *IJCAI Workshop on Autonomous Mobile Service Robots*.
- [171] Sun Wu, Udi Manber, Gene Myers, and Webb Miller. 1990. An O (NP) sequence comparison algorithm. *Inform. Process. Lett.* 35, 6 (1990), 317–323.
- [172] Fabio Massimo Zanzotto. 2019. Human-in-the-loop Artificial Intelligence. *Journal of Artificial Intelligence Research* 64 (2019), 243–252.
- [173] V. A. Ziparo, L. Iocchi, D. Nardi, P. F. Palamara, and H. Costelha. 2008. Petri Net Plans: A Formal Model for Representation and Execution of Multi-robot Plans. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1 (AAMAS ’08)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 79–86.
- [174] Norbert Zmyj, Moritz M Daum, and Gisa Aschersleben. 2009. The development of rational imitation in 9- and 12-month-old infants. *Infancy* 14, 1 (2009), 131–141.

Appendix A Full Results from the Studies in Desired Robot Explanation

A.1 Internal Consistency (2019 Study vs 2020 Study)

Table 9: Explanation Measure Items

| |
|--|
| Unexpectedness (Cronbach's $\alpha = 0.80$ vs. 0.70) <i>1. I found the robot's behavior confusing.</i> <i>2. The robot's behavior matched what I expected. (Reversed)</i> <i>3. The robot's behavior surprised me.</i> |
| Need (Cronbach's $\alpha = 0.74$ vs. 0.65) <i>1. I want the robot to explain its behavior.</i> <i>2. The robot should not explain anything about its behavior. (Reversed)</i> |
| Human-Robot Difference (Cronbach's $\alpha = 0.49$ vs. 0.60) <i>1. There should be no difference between what a robot says to explain its behavior and what a person would say to explain the same behavior.</i> <i>2. If a person did what the robot did, they should both explain the same behavior in the same way.</i> |
| Summarization (Cronbach's $\alpha = -0.57$ vs. -0.77 ; 0.65 vs. 0.56 if Q1 is dropped) <i>1. The robot should give a very detailed explanation. (Reversed)</i> <i>2. The robot should concisely explain its behavior.</i> <i>3. The robot should give a summary about its behavior before giving more detail.</i> |

* Likert items are coded as -3 (Strongly Disagree), -2 (Disagree), -1 (Moderately Disagree), 0 (Neutral), 1 (Moderately Agree), 2 (Agree), and 3 (Strongly Agree).

Cronbach's α for Need dropped from 0.74 to 0.65. The reason is possibly that the second Need question is phrased as a double negative and difficult to understand. To answer, it becomes "I (strongly/moderately) disagree that the robot should not explain anything about its behavior". We plotted the responses to both questions and found that they do not belong to the same distribution. Specifically, responses to the second question are not normally distributed. We thus analyzed the first Need question and present the results in Appendix A.3.

A.2 Unexpectedness

A.2.1 Unexpectedness (interaction effect): No change

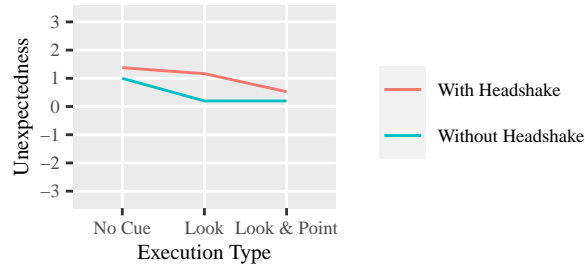


Figure 61: Interaction plot of the unexpectedness responses (original result from the 2019 Study).

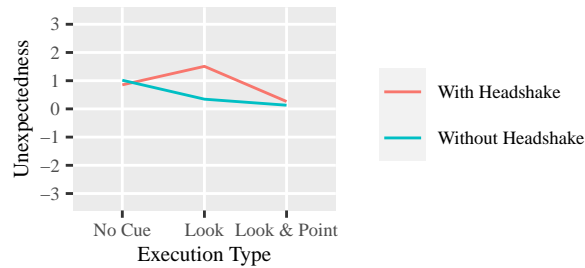


Figure 62: Interaction plot of the unexpectedness responses (replication result from the 2020 Study).

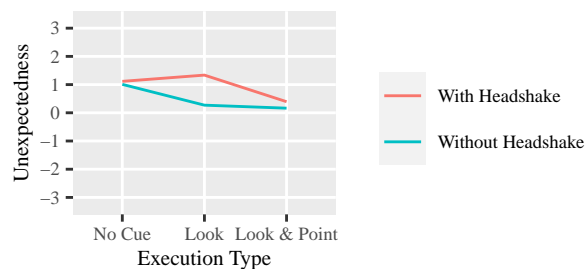


Figure 63: Interaction plot of the unexpectedness responses (combined result).

In the 2019 Study, no interaction effect was suggested by a two-way between-subjects factorial ANOVA. In the replication (2020 Study), the same ANOVA test suggested there might be an

interaction effect ($p < 0.001$) but post-hoc pairwise comparisons using Tukey's test with Holm-Bonferroni correction did not find a statistical significance between No Cue conditions (leftmost column in Figure. 65).

In terms of main effects in the 2019 Study, a two-way between-subjects factorial ANOVA revealed 2 statistically significant main effects in the original experiment for both *Headshake* ($F(1, 360) = 15.30, p < 0.001$) and *Execution Type* ($F(2, 360) = 11.98, p < 0.0001$).

In the 2020 Study, main effects were found again for *Headshake* ($F(2, 360) = 7.07, p < 0.001$) and *Execution Type* ($F(2, 360) = 11.90, p < 0.0001$).

A.2.2 Unexpectedness (bar chart)

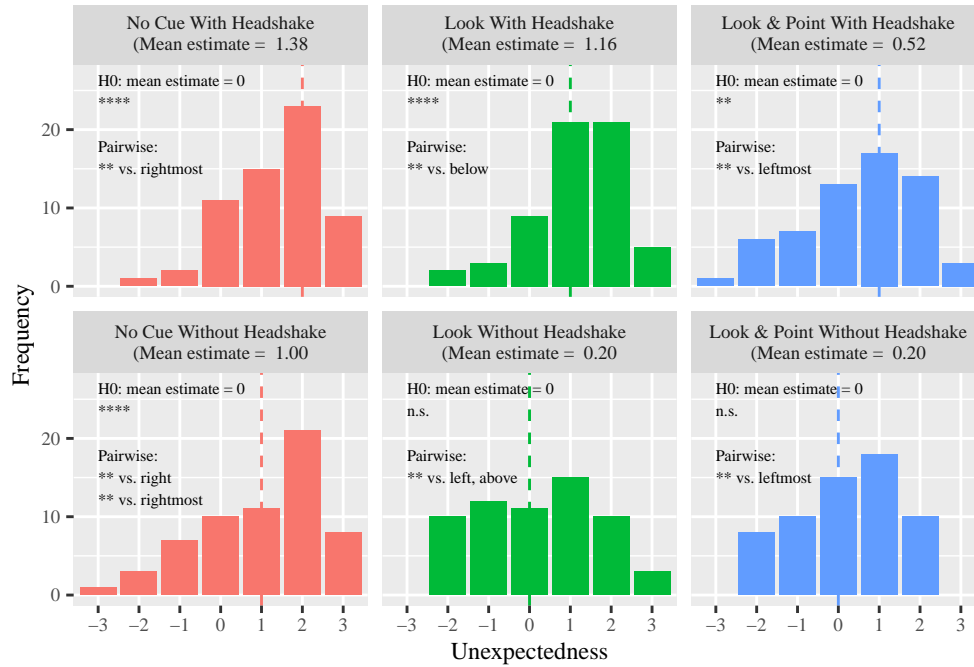


Figure 64: The distribution of Unexpectedness responses with median lines (original result from the 2019 Study).

Comparing Figure 64 and Figure 65, Look & Point with Headshake (top right) changed for $H_0 : \mu = 0$, meaning participants neither disagreed nor agreed that the Look & Point with Headshake is unexpected. This statistically significant change led to pairwise comparison changes for the condition ($H_0 : \mu_i = \mu_j$), in addition to pairwise comparison changes for the conditions in the left 2 columns.

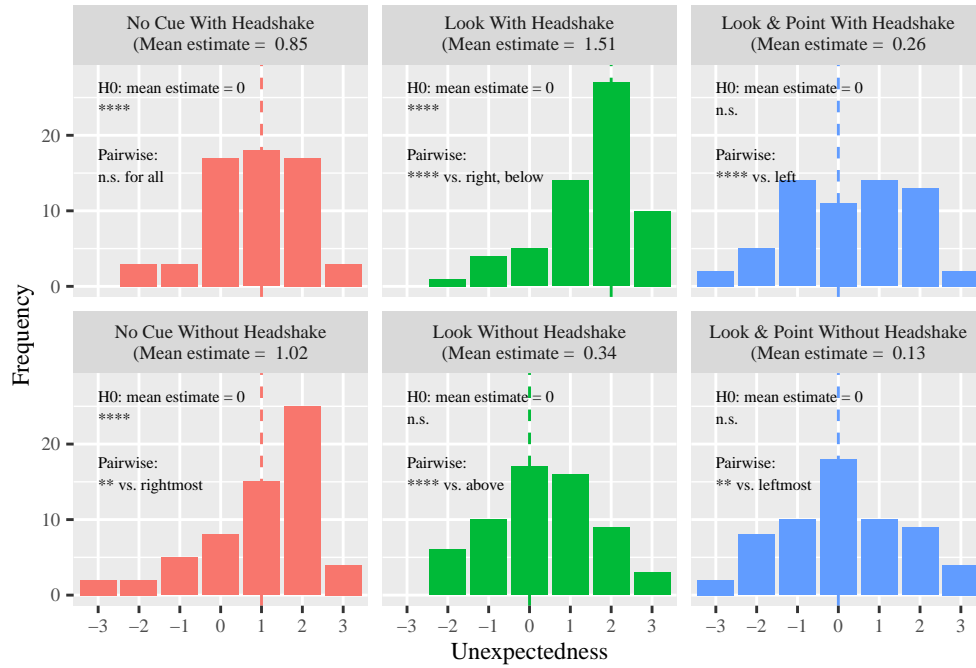


Figure 65: The distribution of Unexpectedness responses with median lines (replication result from the 2020 Study).

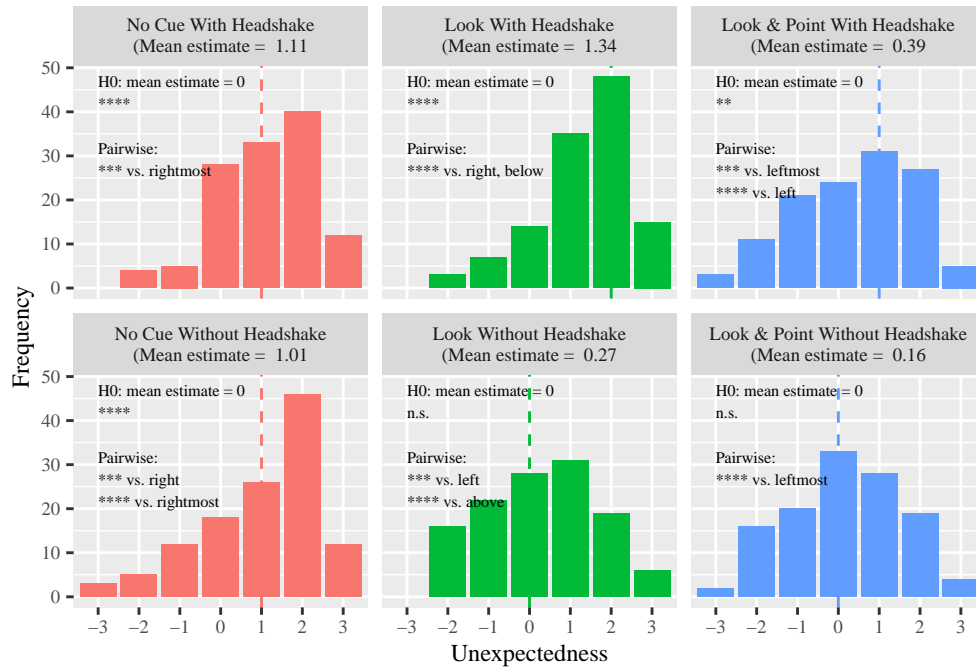


Figure 66: The distribution of Unexpectedness responses with median lines (combined result).

A.2.3 Unexpectedness (box plot)

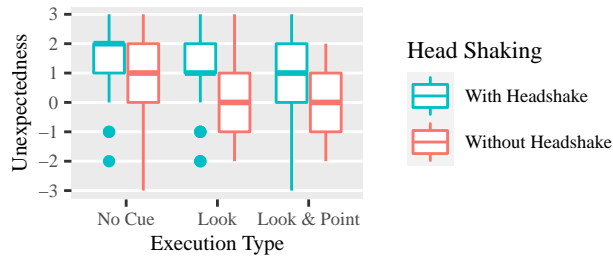


Figure 67: Boxplot of Unexpectedness responses (original result from the 2019 Study).

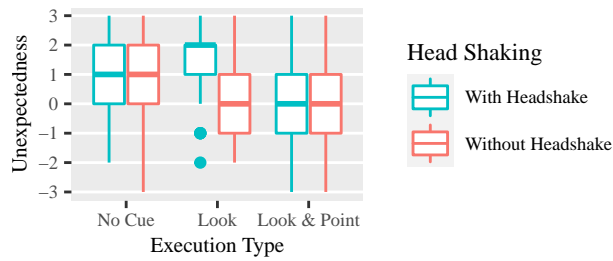


Figure 68: Boxplot of Unexpectedness responses (replication result from the 2020 Study).

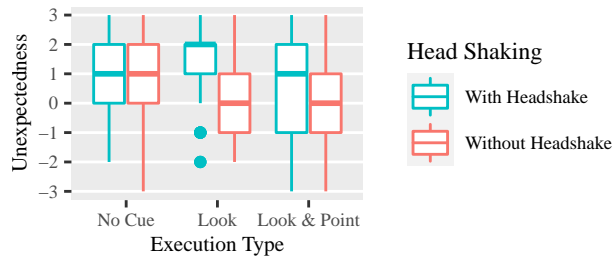


Figure 69: Boxplot of Unexpectedness responses (combined result).

A.3 Need (Question 1): Same conclusions

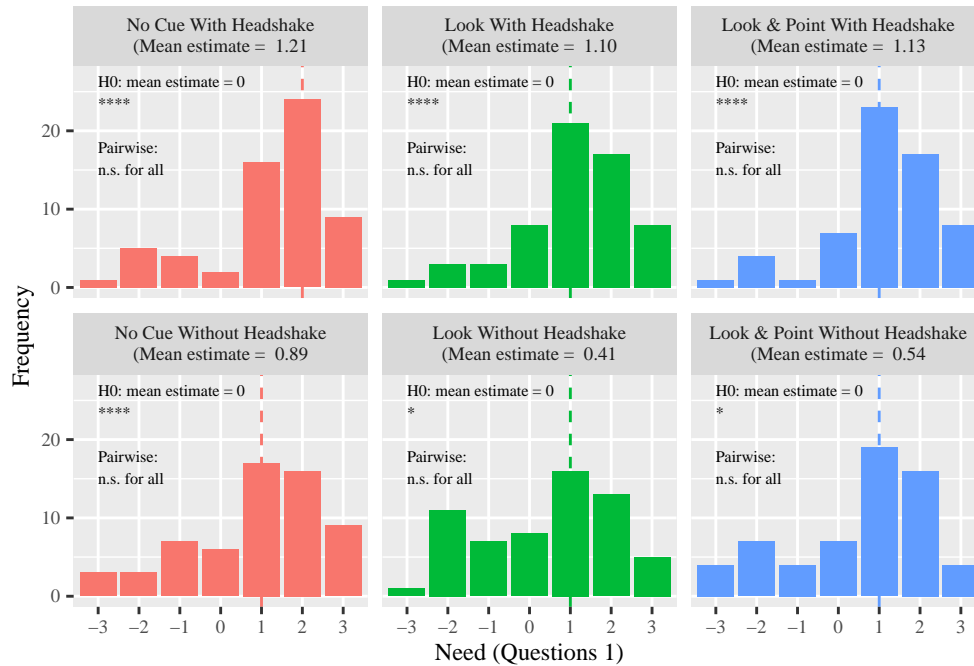


Figure 70: The distribution of Need (questions 1) responses with median lines (original result from the 2019 Study).

The conclusion about the need to explain still holds. To analyze the results, we used the same significance tests as the ones in Section 3.4.2.

A.3.1 Results for 2019 Study (Figure 70)

A between-subjects factorial ANOVA did not show a statistically significant interaction between *Headshake* and *Execution Type*, but found a statistically significant main effects for *Headshake* ($F(1, 360) = 11.10, p < 0.001$) but not *Execution Type* for explanation scores.

Before we conducted pairwise comparisons across conditions, we used the ANOVA model to calculate estimated marginal means of all conditions and performed multiple comparisons with Holm-Bonferroni correction [89] to test whether these means significantly deviate from 0 ($H_0 : \mu = 0$). Results showed statistical significance across all conditions:

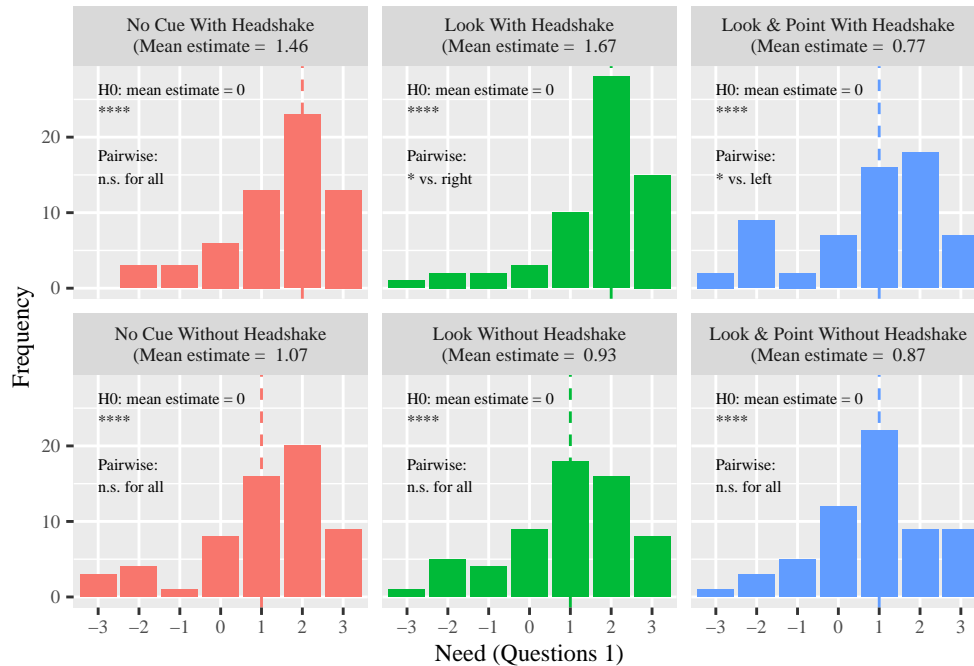


Figure 71: The distribution of Need (questions 1) responses with median lines (replication result from the 2020 Study).

- Without headshake:
 - Look & Point: 0.54 ± 0.20 , $p < 0.0001$
 - Look: 0.41 ± 0.20 , $p < 0.001$
 - No Cue: 0.89 ± 0.20 , $p < 0.0001$
- With headshake:
 - Look & Point: 1.13 ± 0.20 , $p < 0.0001$
 - Look: 1.10 ± 0.20 , $p < 0.0001$
 - No Cue: 1.21 ± 0.20 , $p < 0.0001$

Pairwise comparisons were conducted with Tukey's test ($H_0 : \mu_i = \mu_j$), not revealing any statistically significant differences between the Execution Type conditions.

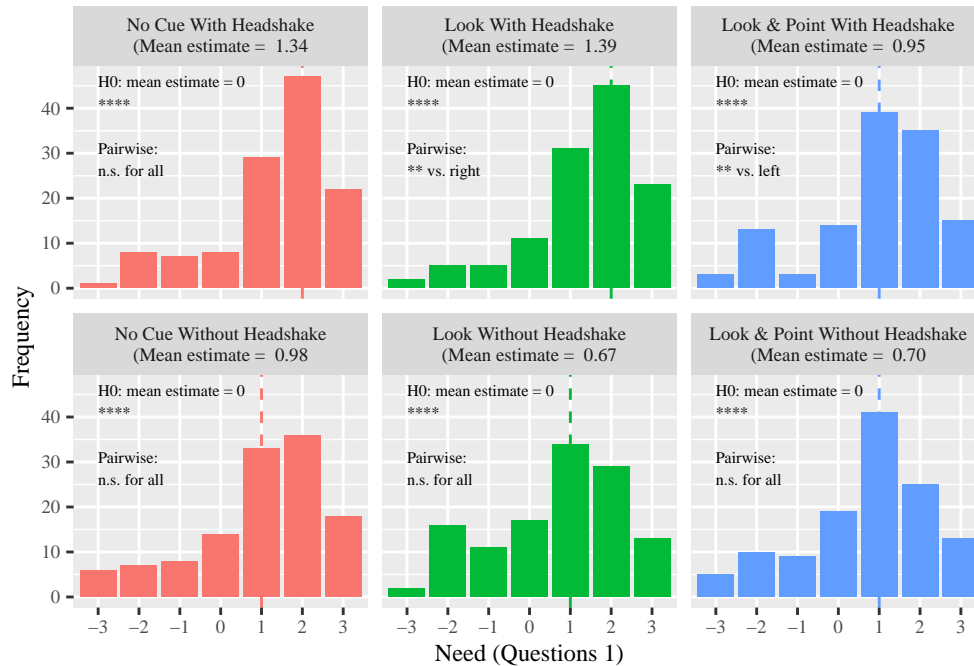


Figure 72: The distribution of Need (questions 1) responses with median lines (combined result).

A.3.2 Results for 2020 Study (Figure 71)

A between-subjects factorial ANOVA did not show a statistically significant interaction between *Headshake* and *Execution Type*, but found statistically significant main effects for *Headshake* ($F(1, 360) = 4.96, p < 0.05$) and *Execution Type* ($F(2, 360) = 4.01, p < 0.05$) for explanation scores.

Before we conducted pairwise comparisons across conditions, we used the ANOVA model to calculate estimated marginal means of all conditions and performed multiple comparisons with Holm-Bonferroni correction [89] to test whether these means significantly deviate from 0 ($H_0 : \mu = 0$). Results showed statistical significance across all conditions ($p < 0.0001$):

- Without headshake:
 - Look & Point: $0.87 \pm 0.19, p < 0.0001$
 - Look: $0.93 \pm 0.19, p < 0.001$

- No Cue: $1.07 \pm 0.19, p < 0.0001$
- With headshake:
 - Look & Point: $0.77 \pm 0.19, p < 0.0001$
 - Look: $1.67 \pm 0.19, p < 0.0001$
 - No Cue: $1.46 \pm 0.19, p < 0.0001$

Pairwise comparisons were conducted with Tukey's test ($H_0 : \mu_i = \mu_j$), revealing only a $p < 0.05$ statistically significant difference between Look with Headshake and Look & Point with Headshake.

A.3.3 Results for both experiments combined (Figure 72)

A between-subjects factorial ANOVA did not show a statistically significant interaction between *Headshake* and *Execution Type*, but found statistically significant main effects of *Headshake* ($F(1, 360) = 16.10, p < 0.0001$) but not *Execution Type* for explanation scores.

Before we conducted pairwise comparisons across conditions, we used the ANOVA model to calculate estimated marginal means of all conditions and performed multiple comparisons with Holm-Bonferroni correction [89] to test whether these means significantly deviate from 0 ($H_0 : \mu = 0$). Results showed statistical significance across all conditions ($p < 0.001$):

- Without headshake:
 - Look & Point: $0.75 \pm 0.13, p < 0.0001$
 - Look: $0.68 \pm 0.14, p < 0.001$
 - No Cue: $1.00 \pm 0.14, p < 0.0001$
- With headshake:
 - Look & Point: $0.96 \pm 0.13, p < 0.0001$

- Look: $1.37 \pm 0.13, p < 0.0001$
- No Cue: $1.35 \pm 0.13, p < 0.0001$

Pairwise comparisons were conducted with Tukey's test ($H_0 : \mu_i = \mu_j$), not revealing any statistically significant differences between the Execution Type conditions.

A.4 Need: Statistical significance remains the same

A.4.1 Need (interaction effect): No change, still no interaction effect

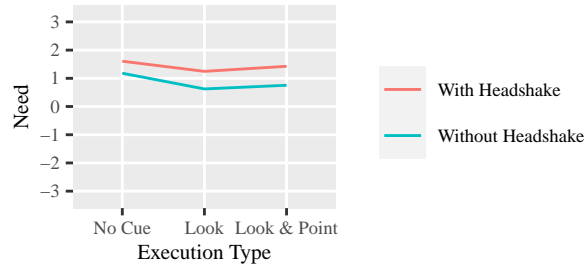


Figure 73: Interaction plot of the unexpectedness responses (original result from the 2019 Study).

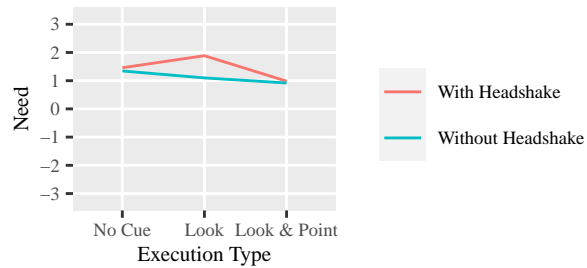


Figure 74: Interaction plot of the Need responses (replication result from the 2020 Study).

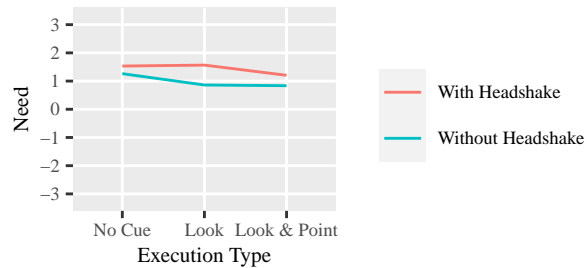


Figure 75: Interaction plot of the Need responses (combined result).

No interaction effect was found in neither the original experiment nor the replication.

For main effects, there were no changes in terms of whether statistical significance was found and just changes significant p values. The main effect of *Headshake* was changed from

$F(1, 360) = 14.99, p < 0.0001$ to $F(1, 360) = 5.32, p < 0.05$. The main effect of *Execution Type* was changed from $F(1, 360) = 3.31, p < 0.05$ to $F(1, 360) = 2.77, p < 0.05$.

A.4.2 Need (bar chart): Same Conclusions

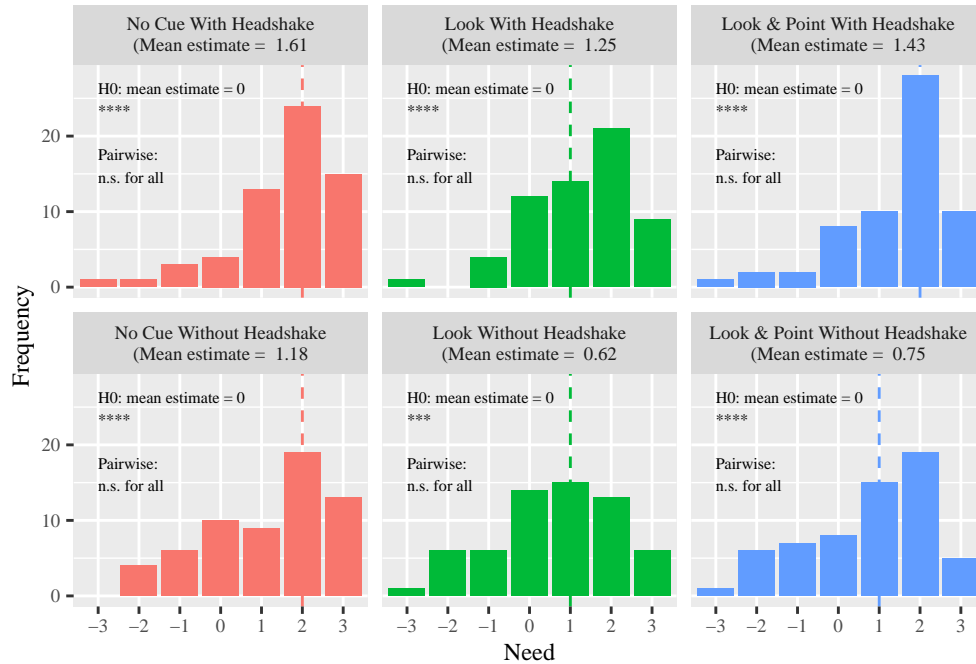


Figure 76: The distribution of Need responses with median lines (original result from the 2019 Study).

There were no changes for $H_0 : \mu = 0$, meaning participants still agreed that the robot should explain across all conditions, we found statistically significant differences between 2 pairs in pairwise comparisons – still remaining to agree. They were between Look with Headshake and Look & Point with Headshake ($p < 0.01$) and Look with Headshake and Look without Headshake ($p < 0.05$).

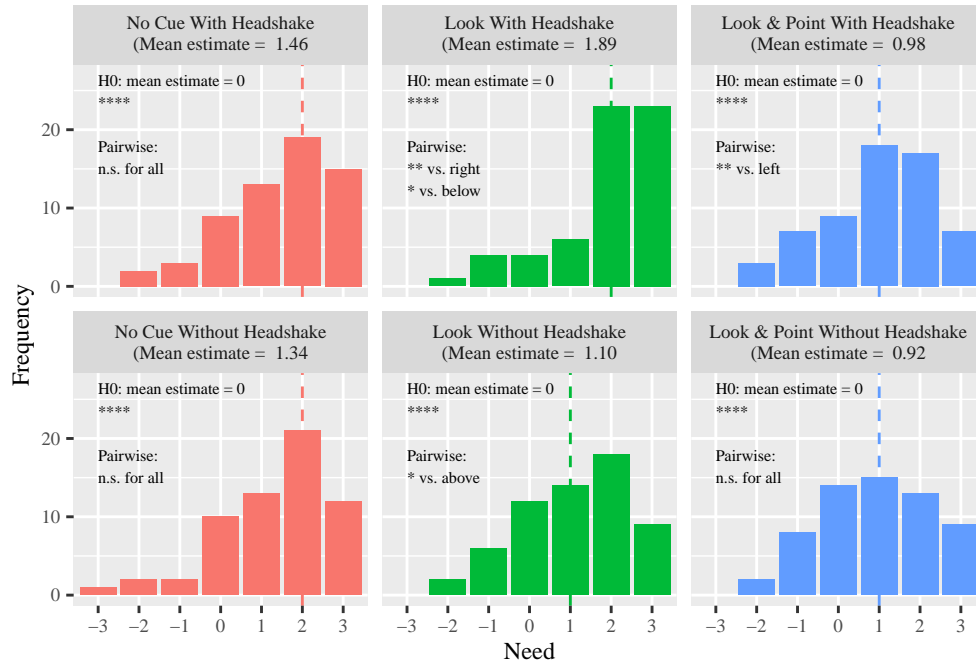


Figure 77: The distribution of Need responses with median lines (replication result from the 2020 Study).



Figure 78: The distribution of Need responses with median lines (combined result).

A.4.3 Need (box plot)

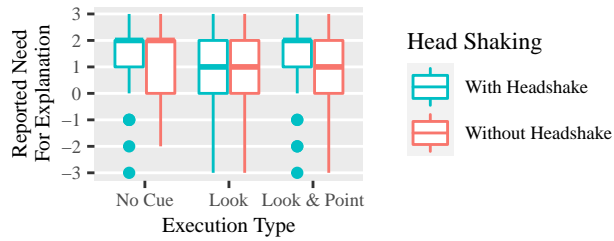


Figure 79: Boxplot of the Need for explanation responses (original result from the 2019 Study).

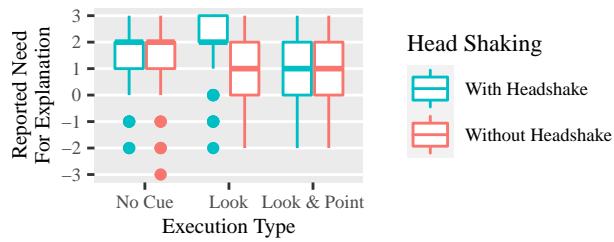


Figure 80: Boxplot of the Need for explanation responses (replication result from the 2020 Study).

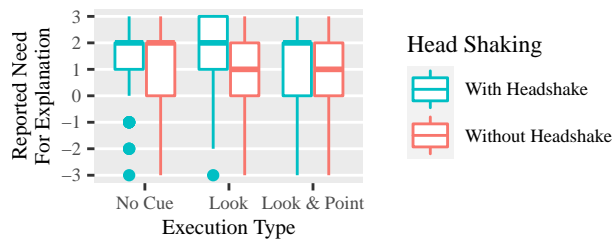


Figure 81: Boxplot of the Need for explanation responses (combined result).

A.5 Expected properties

A.5.1 Explanation timing and verbosity preferences: Conclusions remain the same

Timing: “A priori” changed from *** to n.s. This has not changed our conclusion that participants wanted robots to explain in situ, not at the end of the task. Verbosity: results remained the same.

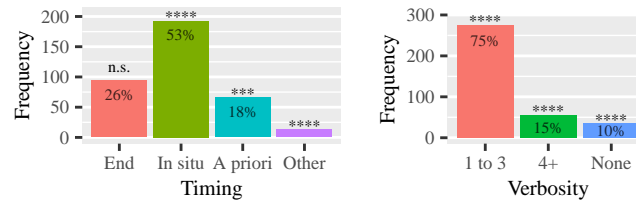


Figure 82: Timing and verbosity preferences (original result from the 2019 Study).

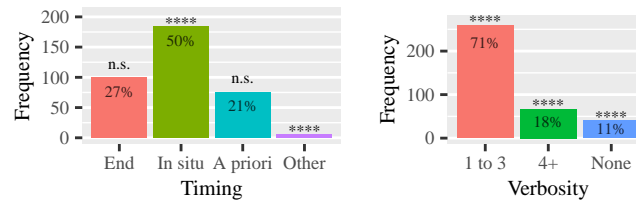


Figure 83: Timing and verbosity preferences (replication result from the 2020 Study).

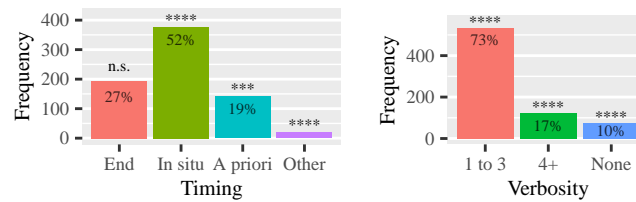


Figure 84: Timing and verbosity preferences (combined result).

A.5.2 Engagement importance/preference: Conclusions remain the same

Importance: no change in terms of statistical significance and significance levels. Preference: the conclusion remains the same but fewer participants chose “Other”.

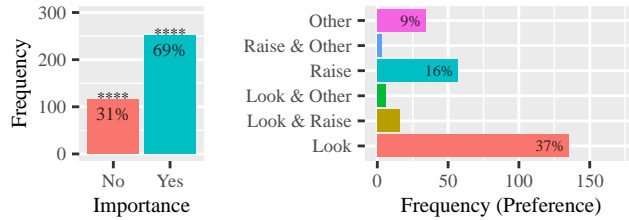


Figure 85: Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). [original result from the 2019 Study]

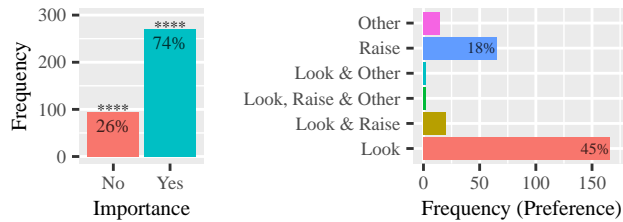


Figure 86: Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). [replication result from the 2020 Study]

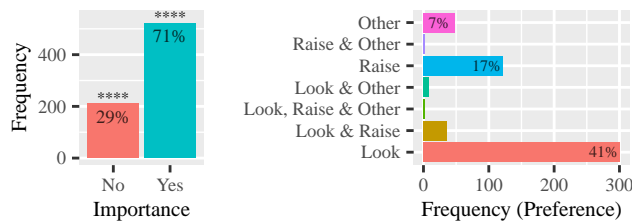


Figure 87: Engagement (Look: Look at me, Raise: Raise volume, Other: Other (Please elaborate)). [combined result]

A.5.3 Similarity to human explanation: Largely remained the same; the same conclusion

What: Scale point 2 moved from **** to ***, 3 from **** to *. How: scale point -1 moved from n.s. to *, 2 from n.s. to ****, 3 from ** to n.s.

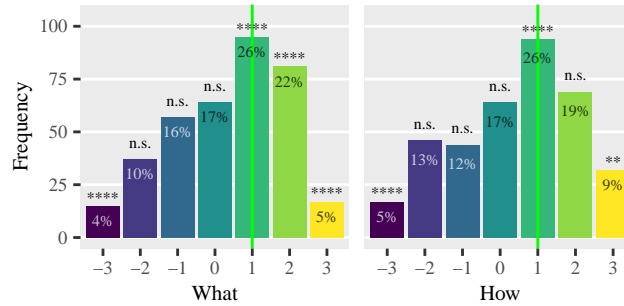


Figure 88: Robot vs. human explanations in what and how (green line indicates median). [original result from the 2019 Study]

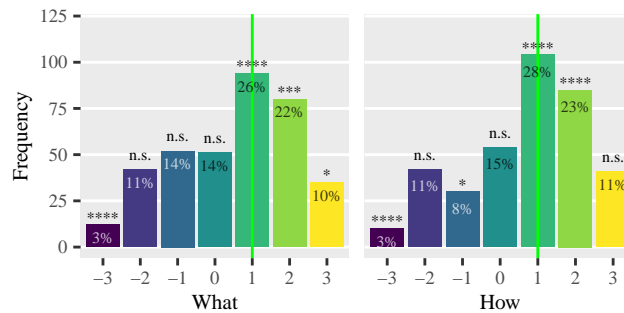


Figure 89: Robot vs. human explanations in what and how (green line indicates median). [replication result from the 2020 Study]

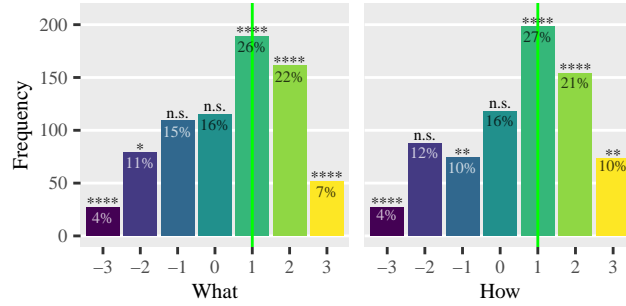


Figure 90: Robot vs. human explanations in what and how (green line indicates median). [combined result]

A.5.4 Detail, summarization: Same conclusions

Detailed: scale point -2 moved from n.s. to **, 2 from n.s. to *, 3 from **** to n.s. Summarized: scale point moved 3 from **** to **.

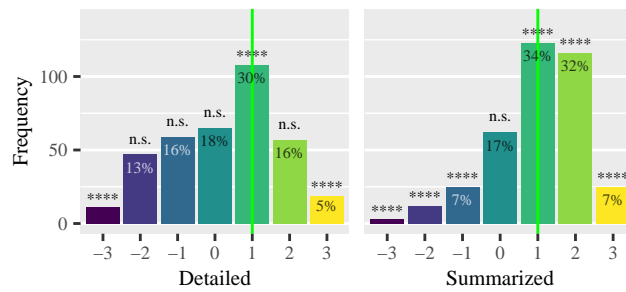


Figure 91: Two summarization aspects (green lines indicate median values). [original result from the 2019 Study]

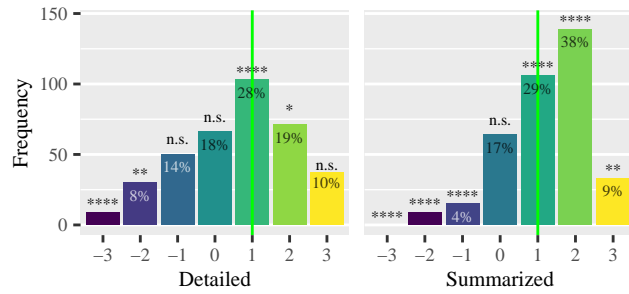


Figure 92: Two summarization aspects (green lines indicate median values). [replication result from the 2020 Study]

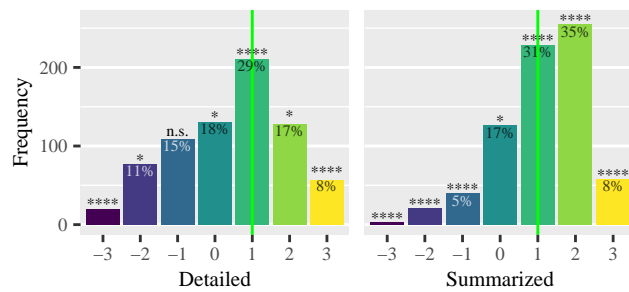


Figure 93: Two summarization aspects (green lines indicate median values). [combined result]