UNIVERSITY OF CALIFORNIA
Los Angeles

Learning How and Why:
Causal Learning and Explanation from
Physical, Interactive, and Communicative Environments

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Mark Joseph Edmonds

2021

ABSTRACT OF THE DISSERTATION

Learning How and Why:
Causal Learning and Explanation from
Physical, Interactive, and Communicative Environments

by

Mark Joseph Edmonds
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2021
Professor Song-Chun Zhu, Chair

Artificial agents expected to operate alongside humans in daily life will be expected to handle novel circumstances and explain their behavior to humans. In this dissertation, we examine these two concepts from the perspective of generalization and explanation. Generalization relies on having a learning algorithm capable of performing well in unseen circumstances and updating the model to handle the novel circumstance. In practice, learning algorithms must be equipped with mechanisms that enable generalization. Here, we examine the generalization question from multiple perspectives, namely imitation learning and causal learning. We show that generalization performance benefits from understanding abstract high-level task structure and low-level perceptual inductive biases. We also examine explanations in imitation learning and communicative learning paradigms. These explanations are intended to foster human trust and address the value alignment problem between humans and machines. In the imitation learning setting, we show that the model components that best contribute to fostering human trust do not necessarily correspond to the model components contributing

most to task performance. In the communicative learning paradigm, we show how theory of mind can align a machine's values to the preferences of a human user. Taken together, this dissertation helps address two of the most critical problems facing AI systems today: machine performance in unseen scenarios and human-machine trust.

The dissertation of Mark Joseph Edmonds is approved.

Ying Nian Wu

Demetri Terzopoulos

Hongjing Lu

Guy Van den Broeck

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2021

Dedicated to every aspiring learner on our planet. There is no subject not worth learning and no idea not worth pondering. Keep exploring.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Family, friends, advisers, collaborators, and labmates; I could not have done this without you. In particular, thank you to:

My parents and Alexandra for their unending support and understanding of my Ph.D. journey. I could not have done it without you.

Professor Song-Chun Zhu for the endless support, believing in me, and encouraging everyone in VCLA to pursue hard, long-term challenges in AI.

Professor Hongjing Lu for teaching me the ways of cognitive science and cultivating my fascination with human intelligence.

Professor Ying Nian Wu for being a constant source of support, theoretical insight, and thoughtful conversation.

Professors Demetri Terzopoulos and Guy Van den Broeck for their time and support on my committee.

Professor Yixin Zhu for being my "big brother" in the lab and lifelong friendship.

Siyuan Qi, Feng Gao, Xu Xie, Hangxin Liu, Chi Zhang, Baoxiong Jia, Yuxing Qiu, Shuwen Qiu, Sirui Xie, Xiaojian Ma, and everyone in the "robot room" in the lab. Special thanks to all VCLA members for their friendship and support over the years.

The dissertation text is a combination of multiple published papers. Chapter 2 is a version of [EGL19], published in *Science Robotics* in 2019. It also contains material published in [EGX17, LXM17]. The author contributions are the following: Mark Edmonds contributed to data collection, grammar induction, haptic network, and explanation interface design; Feng Gao contributed to robot deployment, grammar induction, and haptic network; Hangxin Liu contributed to the glove design, data collection, and explanation interface design; Xu Xie contributed to the glove design, data collection, and haptic network; Siyuan Qi contributed to the GEP integration; Brandon Rothrock contributed to data collection,

glove design, and haptic network; Yixin Zhu contributed to glove design, data collection, and explanation interface design; Ying Nian Wu contributed to the haptic network; Song-Chun Zhu contributed to research direction, data collection and served as the PI.

Chapter 3 is a version of [EMQ20], published in *AAAI* in 2020. It also contains material published in [EKS18, EQZ19]. The author contributions are the following: Mark Edmonds contributed to causal theory induction, experimental design, simulator design, and human subject data collection; Xioajian Ma contributed to baseline experiments; James Kubricht contributed to experimental design and human subject data collection; Colin Summers contributed to simulator design; Siyuan Qi contributed to causal theory induction; Yixin Zhu contributed to causal theory induction, experimental design, and simulator design; Brandon Rothrock contributed to simulator design; Hongjing Lu contribute to experimental design and simulator design; Song-Chun Zhu contributed to research direction and served as the PI.

Chapter 4 is an unpublished chapter, to be submitted as a journal article in the coming months. The author contributions are the following: Mark Edmonds contributed to simulator design, experimental design, and human subject data collection; Luyao Yuan contributed to ToM modeling, experimental design, and human subject data collection; Zilong Zheng contributed to explanation generation and simulator design; Xiaofeng Gao contributed to simulator design and experimental design; Yixin Zhu contributed to simulator design and experimental design; Hongjing Lu contributed to experimental design; Song-Chun Zhu contributed to research direction and served as the PI.

This work was supported by DARPA XAI N66001-17-2-4029, ONR MURI N00014-16-1-2007, and DARPA SIMPLEX N66001-15-C-4035. I am grateful for the collaboration, inspiration, and funding support these grants provided.

VITA

2016–2021	Graduate Student Researcher, UCLA.
2016–2021	Adjunct Professor, Santa Monica College.
2019	Ph.D. Candidate (Computer Science), UCLA).
2017	M.S. (Computer Science), UCLA.
2015–2016	Teaching Assistant, Computer Science Department, UCLA.
2015	B.S. (Computer Engineering), University of Dayton.
2015	The Anthony Horvath and Elmer Steger Award of Excellence, University of Dayton.
2014	Tau Beta Pi Engineering Honor Society, Member.
2014	Eta Kappa Nu IEEE Honor Society, Member.

PUBLICATIONS

$\star$ denotes joint first authors

ACRE: Abstract Causal REasoning Beyond Covariation. C. Zhang, B. Jia, **M. Edmonds**, S.C. Zhu Y. Zhu. *CVPR* 2021.

Dark, Beyond Deep: A Paradigm Shift to Cognitive AI with Humanlike Common Sense. Y. Zhu, T. Gao, L. Fan, S. Huang, **M. Edmonds**, H. Liu, F. Gao, C. Zhang, S. Qi, Y.N. Wu, J.B. Tenenbaum, S.C. Zhu. *Engineering*, 2020.

Theory-based Causal Transfer: Integrating Instance-level Induction and Abstract-level Structure Learning. **M. Edmonds**, X. Ma, S. Qi, Y. Zhu, H. Lu, S.C. Zhu. *AAAI*, 2020.

A tale of two explanations: Enhancing human trust by explaining robot behavior. **M. Edmonds***, F. Gao*, H. Liu*, X. Xie*, S. Qi, B. Rothrock, Y. Zhu, Y.N. Wu, H. Lu, S.C. Zhu. *Science Robotics*, 2019.

Decomposing Human Causal Learning: Bottom-up Associative Learning and Top-down Schema Reasoning. **M. Edmonds**, S. Qi, Y. Zhu, J. Kubricht, S.C. Zhu, H. Lu. *CogSci*, 2019.

Human Causal Transfer: Challenges for Deep Reinforcement Learning. **M. Edmonds***, J. Kubricht*, C. Summers, Y. Zhu, B. Rothrock, S.C. Zhu, H. Lu. *CogSci*, 2018.

Unsupervised Learning of Hierarchical Models for Hand-Object Interactions. X. Xie*, H. Liu*, **M. Edmonds**, F. Gao, S. Qi, Y. Zhu, B. Rothrock, S.C. Zhu. *ICRA*, 2018.

Feeling the Force: Integrating Force and Pose for Fluent Discovery through Imitation Learning to Open Medicine Bottles. **M. Edmonds***, F. Gao*, X. Xie, H. Liu, S. Qi, Y. Zhu, B. Rothrock, S.-C. Zhu. *IROS*, 2017.

A Glove-based System for Studying Hand-Object Manipulation via Pose and Force Sensing. H. Liu*, X. Xie*, M. Millar*, **M. Edmonds**, F. Gao, Y. Zhu, V. Santos, B. Rothrock, S.C. Zhu. *IROS*, 2017.

CHAPTER 1

Introduction

Generalization, transfer, and explanation are critical abilities for artificial general intelligence (AGI). Humans are able to generalize rapidly and apply domain knowledge from one area to another. This transfer can be applying from one area of expertise in a domain to another (near transfer) or from one domain to a completely different domain (far transfer, analogical reasoning). These types of transfer always rely on forming some notion of a prior belief about knowledge within a domain and applying that prior in a useful way in another domain.

Humans are apt at taking knowledge from one domain and applying it to another. Gick, *et al*. present a classic example of analogical transfer with human subjects [GH80]. Subjects are first informed about a radiation problem [DL45], where a patient must have a tumor destroyed using a specific type of ray. At high intensities, a ray will destroy all human tissue, including healthy tissue. At low intensities, a ray will not destroy healthy or cancerous tissue. Subjects are then asked how the rays could be used to destroy the cancerous tissue.

Some subjects were first presented with background stories that may help prime them to understand how to solve the problem. For instance, an Attack-Dispersion story was presented to subjects. In the story, a general attempts to capture a fortress, but the attacking army cannot attack at once and must attack in small groups. If the fortress is attacked from multiple sides with small forces, it can be captured (similar to flanking maneuver). This story serves as a prior, and human subjects presented with the story were more likely to generate a solution to the radiation problem. For example, subjects may argue that multiple low-intensity rays might meet at the tumor and constructively interfere to an intensity that

can destroy the tumor but leave healthy human tissue intact.

Performing this mapping from one domain to another requires understanding a few different concepts. First, one must know the structural properties of one domain to map it onto another. For instance, it is a prerequisite to understand why sending small groups of soldiers from multiple directions is an effective strategy in battle to understand how the knowledge can be applied in the radiation problem. Without knowledge in the source domain, it is not possible to apply a transformation to transform knowledge in the source domain to a new domain. Priors are a necessary part of knowledge transfer; with no prior knowledge, there is no knowledge to transfer to a new domain.

Second, the agent must have a mechanism by which to construct a mapping from one domain to another. There are a few different strategies for creating a mapping. The first would be to directly map concepts in the source domain to the target domain. This works when the concepts neatly align with one another, but finding a perfect or near-perfect mapping is not always feasible. In these settings, concepts and structure in the source domain may need to be combined or reconstructed to map correctly to the target domain. This could be achieved through *abstraction*, where concepts in a source domain are used to form an abstract structure, and then that abstract structure is mapped to the target domain. Understanding both the priors necessary and structural abstraction necessary is crucial to solve generalization tasks. This dissertation will look at three settings to explore these issues, outlined below.

Generalization and explanation in observational domains: Chapter 2 examines an observational setting where an agent imitates a demonstration to achieve a task and is then tasked with generalizing their knowledge to similar but unseen circumstances. In this setting, a robot learns how to manipulate a medicine bottle by imitating a human demonstrator. We will show that the robot effectively learns a policy from observations that generalizes well to unseen bottles using a top-down and bottom-up approach. The top-down modeling

component encodes the long-term, symbolic task structure in the form of a temporal And-Or graph (T-AOG) while the bottom-up component allows the robot to imagine itself as a human demonstrator and predict what the human would have done next under similar poses and forces.

In this section, we will also explore an idea related to generalization - explanation. Of the space of possible representations, the representations that generalize are likely to also contain some explanatory power. If learned knowledge generalizes well, it must capture some fundamental portions of the underlying task, and therefore, must contain some explanatory power around those fundamental task components. In the bottle opening task, the robot provides explanations into how it reasoned about the medicine bottle, and human observers rate how much they trust the robot under different explanatory formats. Participants are also asked to predict what they think the robot will do next, thereby examining the ability of an explanatory format to impart a user with the ability to predict future actions by the machine. The results show that the modeling components that best contribute to task performance are not necessarily the modeling components that foster the most trust, indicating a need to consider both task performance and explanation as separate but critical components for robots interacting with humans.

Causal generalization in interactive domains: Chapter 3 showcases the second setting, an interactive setting where agents must interact with their environment and update their understanding of the world based on the outcomes of that interaction. In this domain, the key to knowledge transfer is to explore the space in a way that imparts generalizable knowledge. Specifically, we examine an interventional setting where an agent is placed in a virtual "escape" room. Levers in the room act as a combination to "unlock" the room. After completing a room, the agent will be placed in another room where the levers have been scrambled, but the underlying abstract *pattern* to unlock the room remains the same. In this setting, the key to generalization is to understand the abstract structure. Once the

structure is known, an agent faced with a new room simply needs to identify the role of every level in the abstract structure. With the roles identified, an optimal agent can immediately generate the possible ways to unlock the rooms.

In this setting, we examine how well human subjects can solve this task. We find that humans are extremely proficient at identifying the abstract structure and applying it to novel domains. Additionally, we examine a plethora of model-free RL algorithms and find they are incapable of learning a generalizable policy, even under favorable training conditions. Finally, we present a hierarchical Bayesian learner that uses abstract top-down structure knowledge to learn a prior over the space of possible causal structures and bottom-up feature knowledge to learn which features may provide hints about causal components in the scene. The Bayesian learner shows similar trends as human learners and achieves near-optimal performance.

Value alignment and explanation in communicative domains: Chapter 4 then expands on the explainable AI work in Chapter 2 by looking at an interactive, communicative learning setting where each agent has partial information regarding the task. This partial information creates the need for explanation (communication) between the agents. In the setting, a human user is overseeing a group of scouts navigate in a dangerous area. The human user understands the values of the group, and the scouts relay information (observations) to the human. Thus the communication facilitates a *value alignment* problem, where the robot scouts must align their values to human values based on feedback from the human. The results show the scouts are able to align successfully with human values, and at a given time point, humans are able to infer the current value function of the scouts (even when it is different from their own). These results are promising for solving value alignment problems with a communicative learning framework.

Together, these three settings encompass important general cases of knowledge transfer and explanation: one where an agent can only rely on observations to learn a generalizable policy, another where the agent must learn how to generalize by actively intervening in

the environment, and a third where agents with partial information must communicate and collaborate to achieve a task. We believe that learning from imitation, intervention, and communication have fundamentally different properties for generalization. From observations, a learner can at best identify causal connections up to a certain degree, though this does not inhibit a learner from learning a policy that can generalize well to unseen environments. From interventions, one can effectively learn causal relationships through deliberate experimentation, though a learner needs effective priors to guide the learning process. Without such priors, the space of possible causal relations is exponential and too vast to explore effectively. From communication and explanation, agents can align values and act in accordance with each other's preferences and beliefs.

This dissertation explores and attempts to answer some of the most pressing issues for current machine learning systems; generalization, transfer, and explanation are all unsolved problems that pose critical and challenging problems for the future of AI. Here, we attempt to add clarity to the conversation by exploring different settings to highlight the importance and difficulty of answering these questions. We advance the state of the art in imitation learning, casual learning, explainable AI, and communicative learning to achieve these goals. The fundamental questions moving forward are how this work can be expanded and scaled to larger experiments and how we can integrate these ideas to create a capable learner and explainer in a wide variety of settings.

CHAPTER 2

Generalization and Explanation in Imitation Learning

In this chapter, we examine generalization in an imitation learning setting and show a robot capable of transferring knowledge to novel problems using a high-level task planner and a low-level haptic action predictor. Consider the task of opening medicine bottles that have child-safety locking mechanisms (Fig. 2.1(a)). These bottles require the user to push or squeeze in various places to unlock the cap. By design, attempts to open these bottles using a standard procedure will result in failure. Even if the agent visually observes a successful demonstration, imitation of this procedure will likely omit critical steps in the procedure. The visual procedure for opening both medicine and traditional bottles is typically identical. The agent lacks understanding of the tactile interaction required to unlock the safety mechanism of the bottle. Only direct observation of forces or instruction can elucidate the correct procedure (Fig. 2.1(e)). Even with knowledge of the correct procedure, opening medicine bottles poses several manipulation challenges that involve feeling and reacting to the internal mechanisms of the bottle cap. Although the presented study takes opening medicine bottles as an example, many other tasks share similar properties and require non-trivial reasoning such as opening locked doors [SGG08].

This chapter develops an integrated framework consisting of a symbolic action planner using a stochastic grammar as the planner-based representation and a haptic prediction model based on neural networks to form the data-driven representation. In addition to opening medicine bottles, this chapter will examine the explainability of different model components. A hallmark of humans as social animals is the ability to answer this "why" question by

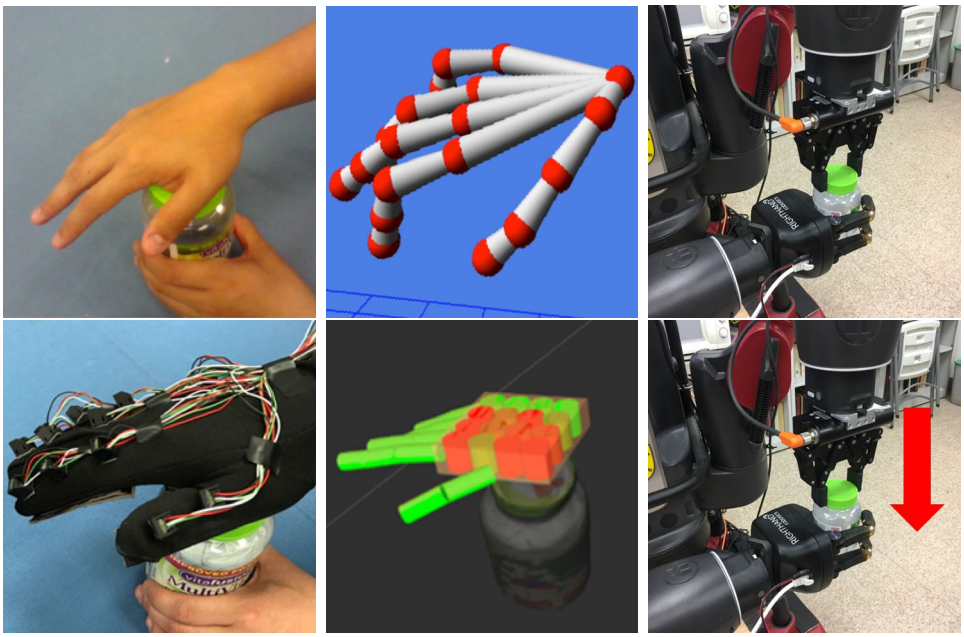


Figure 2.1: Given a RGB-D-based image sequence (a), although we can infer the skeleton of hand using vision-based methods (b), such knowledge cannot be easily transferred to a robot to open a medicine bottle (c), due to the lack of force sensing during human demonstrations. In this chapter, we utilize a tactile glove (d) and reconstruct both forces and poses from human demonstrations (e), enabling robot to directly observe forces used in demonstrations so that the robot can successfully open a medicine bottle (f). Copyright reserved to original publication [EGX17].

providing comprehensive explanations of the behavior of themselves and others. The drive to seek explanations is deeply rooted in human cognition. Preschool-age children tend to attribute functions to all kinds of objects—clocks, lions, clouds, and trees, as explanations of the activity that these objects were apparently designed to perform [Kel99, GMK99]. The strong human preference and intrinsic motivation for explanation are likely due to its central role in promoting mutual understanding, which fosters trust between agents and thereby enables sophisticated collaboration [Lom06, Tom10].

However, a strong human desire for explanations has not been sufficiently recognized

by modern artificial intelligence (AI) systems, in which most methods primarily focus on task performance [Gun17]. Consequently, robot systems are still in their infancy in developing the ability to explain their own behavior when confronting noisy sensory inputs and executing complex multi-step decision processes. Planner-based robot systems can generally provide an interpretable account for their actions to humans (*e.g.*, by Markov decision processes [FHL16, HS17], HTN [EHN96], or STRIPS [FN71]); but these planners struggle to explain how their symbolic-level knowledge is derived from low-level sensory inputs. In contrast, robots equipped with Deep Neural Networks (DNNs) [HOT06] have demonstrated impressive performance in certain specific tasks due to their powerful ability to handle low-level noisy sensory inputs [DCH16, LLS15]. However, DNN-based methods have well-known limitations, notably including a lack of interpretability of the knowledge representation [Mar18, MP17, Dom15]. Some recent DNN work addresses this issue using saliency maps [KRD18, YKY18] or modularized components [HAD18, ZNZ18]. These data-driven approaches have demonstrated strong capabilities of handling noisy real-time sensory inputs, distilling the raw input to predict the effect and determine the next action. However, little work has been done to develop the synergy between the classic symbolic AI and the recent development of DNNs to empower machines with the ability to provide comprehensive explanations of their behavior.

The project in this chapter aims to disentangle explainability from task performance, measuring each separately to gauge the advantages and limitations of two major families of representations—symbolic representations and data-driven representations—in both task performance and imparting trust to humans. The goals are to explore: (i) what constitutes a good performer for a complex robot manipulation task? (ii) How can we construct an effective explainer to explain robot behavior and impart trust to humans?

We examine this integrated framework in a robot system using a contact-rich manipulation task of opening medicine bottles with various safety lock mechanisms. From the performer's perspective, this task is a challenging learning problem involving subtle manip-

ulations, as it requires a robot to push or squeeze the bottle in various places to unlock the cap. At the same time, the task is also challenging for explanation, as visual information alone from a human demonstrator is insufficient to provide an effective explanation. Rather, the contact forces between the agent and the bottle provide the *hidden* "key" to unlock the bottle, and these forces cannot be observed directly from visual input. An overview of the system is shown in Fig. 2.2.

To constitute a good performer, the robot system proposed here cooperatively combines multiple sources of information for high performance, enabling synergy between a high-level symbolic action planner and a low-level haptic prediction model based on sensory inputs. A stochastic grammar model is learned from human demonstrations and serves as a symbolic representation capturing the compositional nature and long-term constraints of a task [TPZ13]. A haptic prediction model is trained using sensory information provided by human demonstrations (*i.e.*, imposed forces and observed human poses) to acquire knowledge of the task. The symbolic planner and haptic model are combined in a principled manner using an improved generalized Earley parser (GEP) [QJZ18], which predicts the next robot action by integrating the high-level symbolic planner with the low-level haptic model. The learning from demonstration framework presented here shares a similar spirit of our previous work [EGX17] but with a new haptic model and a more principled manner, namely the GEP, to integrate the haptic and grammar models. Computational experiments demonstrate a strong performance improvement over the symbolic planner or haptic model alone.

To construct an effective explainer, the proposed approach draws from major types of explanations in human learning and reasoning that may constitute representations to foster trust by promoting mutual understanding between agents. Previous studies suggest humans generate explanations from *functional* perspectives that describe the *effects* or *goals* of actions and from *mechanistic* perspectives that focus on behavior as a process [Lom13]. The haptic prediction model is able to provide a functional explanation by visualizing the essential haptic signals (*i.e.*, *effects* of the previous action) to determine the next action. The

symbolic action planner is capable of providing a mechanistic explanation by visualizing multiple planning steps (instead of just one) to describe the *process* of the task. The proposed robot system provides both functional and mechanistic explanations using the haptic model and symbolic planner, respectively.

To examine how well robot-generated explanations impart human trust, we conduct human experiments to assess whether explanations provided by the robot system can foster trust in human users, and if so, what forms of explanation are the most effective in enhancing human trust in machines. In this chapter, we refer to the cognitive component of "trust" [Sim07] based on rationality. Cognitive trust is especially important in forming trust within secondary groups (such as human-machine relations) [LW85] compared to the emotional component typically more important in primary group relations (such as family and close friends). Our psychological experiment focuses on cognitive trust, stressing on a belief or an evaluation with "good rational reasons," as this is the crucial ingredient of human-machine trust built on specific beliefs and goals with attention to evaluations and expectations [CF98]. Specifically, human participants were asked to report qualitative trust ratings after observing robot action sequences along with different forms of explanations for the robot's internal decision-making as it solved a manipulation task. Then, participants observed similar but new robot executions *without* access to explanations and were asked to predict how the robot system is likely to behave across time. These empirical findings shed light on the importance of learning human-centric models that make the robot system explainable, trustworthy, and predictable to human users. Our results show that forms of explanation that are best suited to impart trust do not necessarily correspond to those components contributing to the best task performance. This divergence shows a need for the robotics community to adopt model components that are more likely to foster human trust and integrate these components with other model components enabling high task performance.

The ideas presented in this chapter were completed across 4 different publications and

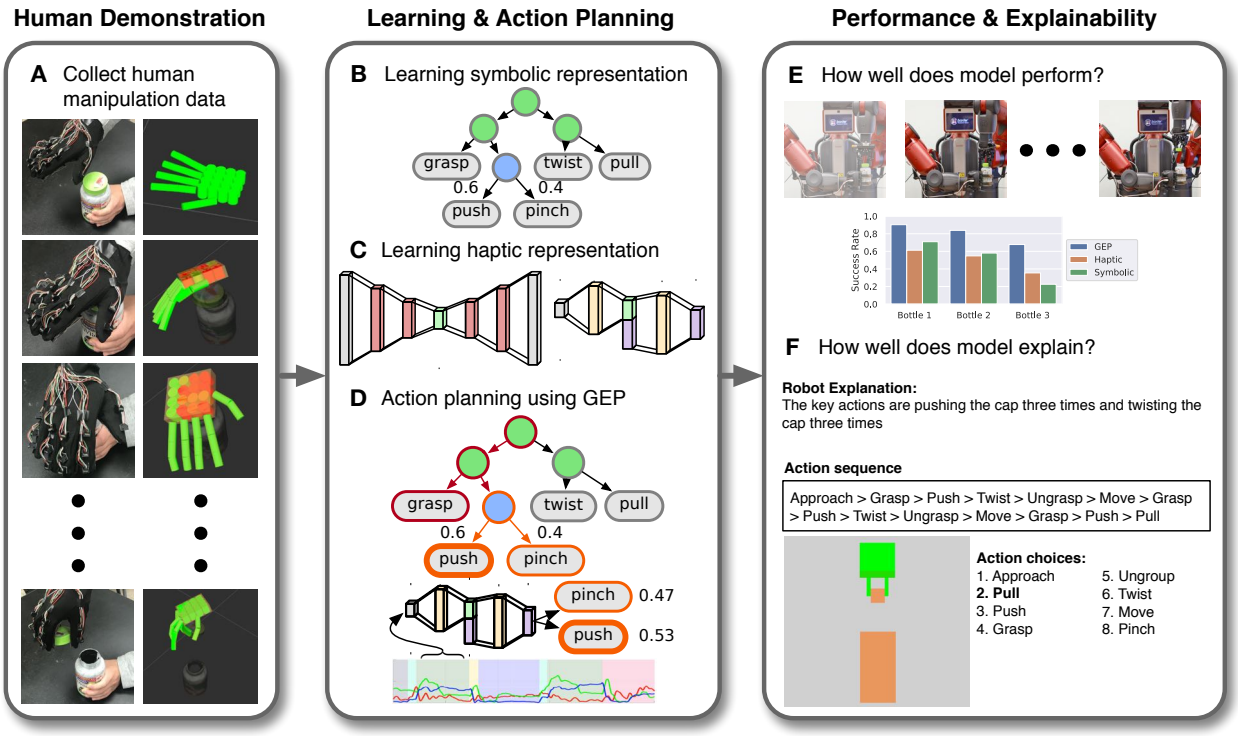


Figure 2.2: Overview of demonstration, learning, evaluation, and explainability. By observing human demonstrations, the robot learns, performs, and explains using both a symbolic representation and a haptic representation. (A) Fine-grained human manipulation data is collected using a tactile glove. Based on the human demonstrations, the model learns (B) symbolic representations by inducing a grammar model that encodes long-term task structure to generate mechanistic explanations, and (C) embodied haptic representations using an autoencoder to bridge the human and robot sensory input in a common space, providing a functional explanation of robot action. These two components are integrated using the (D) GEP for action planning. These processes complement each other in both (E) improving robot performance and (F) generating effective explanations that foster human trust. Copyright reserved to original publication [EGL19].

in collaboration with Feng Gao, Hangxin Liu, Xu Xie, Matt Millar, Siyuan Si, Brandon Rothrock, Veronica Santos, Hongjing Lu, Ying Nian Wu, and Song-Chun Zhu [EGL19,

XLE18, EGX17, LXM17]. The author's contributions to this project include data collection using the tactile glove, learning the T-AOG, training the haptic network, designing and building the explanation interfaces, running the human subject studies, data analysis, and deploying parts of the robot stack (*e.g.* deploying a mobile base system and vision system for the robot). All other portions of the project were not completed by the author. The majority of the material presented in the chapter is copyrighted by the original publisher of [EGL19], and relevant portions have copyrights declared inside the material.

2.1 Tactile Glove and Data Collection

We utilize a tactile glove with force sensor [LXM17] to capture both the poses and the forces involved in human demonstrations in opening medicine bottles that require a visually latent interaction between the hand and the cap, *e.g.*, pushing as indicated in Fig. 2.2A. A human demonstrator performed opening various types of bottles shown in Fig. 2.10A. Some of the bottles contain child-safety locking mechanisms that require a procedure beyond simply twisting to unscrew the cap. Most child-safety locks require a particular force to be exerted on a particular part of the bottle; these forces are difficult to infer from visual observation alone.

Fifteen IMUs obtain the relative poses of finger phalanges with respect to the wrist (see Fig. 2.3A) and develop a customized force sensor using a soft piezoresistive material (Velostat) whose resistance changes under pressure [LXM17]. The 26 force sensors are placed on the palm and fingers, as shown in Fig. 2.3B. The force sensor is constructed in a 5-layer, mirrored structure—Velostat is the inner layer, conductive fabric and wires are the middle layers, and insulated fabric is the outer layer. Fig. 2.3C illustrates the structure of the force sensor, and the force-resistance relation is characterized as Fig. 2.3D. In total, the glove provides 71 degrees of freedom, including all pose and force measurements of the hand, resulting in a fine-grained reconstruction. The relative poses between the hand and manipulating objects

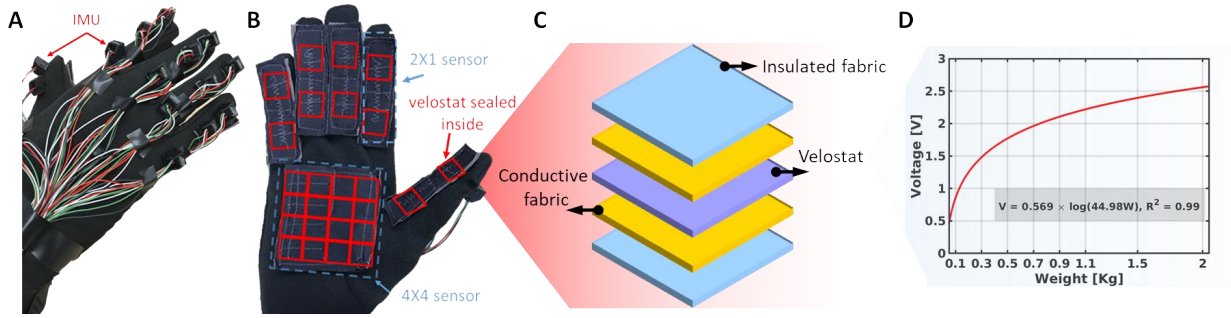


Figure 2.3: (A) The dorsum of the tactile glove developed consisting of 15 IMUs. (B) 26 integrated Velostat force sensor on the palmar aspect of the hand. (C) The structure of the force sensor. (D) Characteristics of the force-voltage relation, which is described by a logarithmic law of the force sensor. Copyright reserved to original publication [EGL19].

(bottles and caps) are captured by a Vicon motion capture system. We captured 64 demonstrations in total; the number of demonstrations varies by the number of possible grasping approaches human demonstrators found natural. Twenty-nine demonstrations were collected for the Bottle 1, 23 for Bottle 2, and 12 for Bottle 3.

The experimental setup is shown in Fig. 2.4. Fiducials are attached to each bottle and its lid to track the pose of object parts. One additional fiducial is attached to the back of the tactile glove to capture wrist pose in world space. A camera is used to record the video of data collection procedures to help label the ground truth later.

A total of 64 human demonstrations, collected in [EGX17], of opening the 3 different medicine bottles serve as the training data. These 3 bottles have different locking mechanisms: no safety lock mechanism, a *push-twist* locking mechanism, and a *pinch-twist* locking mechanism. To test the generalization ability of the robot system, we conduct a generalization experiment with new scenarios different from training data, either a new bottle (Fig. 2.10B) or a bottle with a modified cap with significantly different haptic signals (Fig. 2.11). The locking mechanisms of the bottles in the generalization experiment are similar but not identical (in terms of size, shape, and haptic signals) to the bottles used

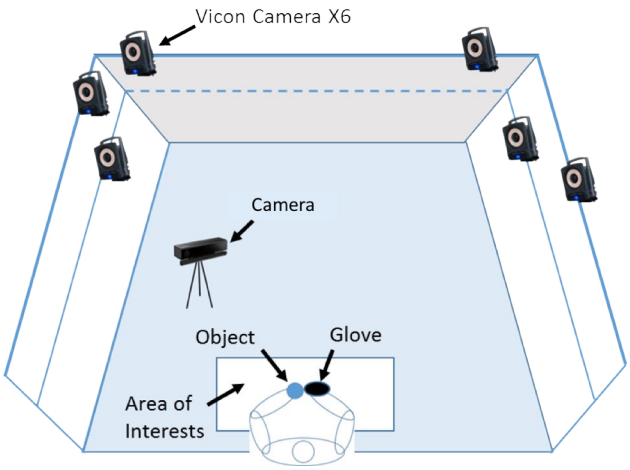


Figure 2.4: We use a Vicon system to obtain the poses of human's wrist and object's parts. The camera is used to record the data collection procedure. Copyright reserved to original publication [EGX17].

in human demonstrations. The haptic signals for the generalization bottles are significantly different from bottles used in testing, posing challenges in transferring the learned knowledge to novel unseen cases.

2.2 Robot Learning

2.2.1 Embodied Haptic Model

Using human demonstrations, the robot learns a manipulation strategy based on the observed poses and forces exerted by human demonstrators. One challenge in learning manipulation policies from human demonstration involves different embodiments between robots and human demonstrators. A human hand has five fingers, whereas a robot gripper may only have two or three fingers; each embodiment exerts different sensory patterns even when performing the very same manipulation. Hence, the embodied haptic model for the robot system cannot simply duplicate human poses and forces exerted by human hands; instead, a robot should imitate the actions with the goal to produce the same end-effect in manipulating the

medicine bottle (*e.g.*, imposing a certain force on the cap). The critical approach in our model is to employ embodied prediction, *i.e.*, let the robot imagine its current haptic state as a human demonstrator and predict what action the demonstrator would have executed under similar circumstances in the next time step. Intuitively, the embodied haptic predictions endow the robot with the ability to ask itself: *if I imagine myself as the human demonstrator, which action would the human have taken next based on the poses and forces exerted by their hand?* Hence, the resulting haptic model provides a *functional* explanation regarding the forces exerted by the robot's actions.

Fig. 2.5 illustrates the force patterns exerted by a robot and a human demonstrator. As shown in panels A and C, due to the differences between a robot gripper and a human hand, the haptic sensing data from robots and humans show very different patterns from each other in terms of dimensionality and duration within each segmented action (illustrated by the colored segments).

To address the cross-embodiment problem, we train a haptic model in a similar approach as in [EGX17] to predict which action the robot should take next based on perceived human and robot forces and poses. The present haptic model learns a prediction model in a three-step process: (i) learning an autoencoder that constructs a low-dimensional embedding of human demonstrations containing poses and forces, as shown in Fig. 2.5B. (ii) Training an embodiment mapping to map robot states to equivalent human embeddings, thereby allowing the robot to imagine itself as a human demonstrator to produce the same force, achieving functional equivalence to generate the same end effect as the human demonstrator. This embodiment mapping is trained in a supervised fashion, using labeled equivalent robot and human states. (iii) Training a next action predictor based on the human embeddings and the current action. This action predictor is also trained in a supervised fashion, using segmented human demonstrations.

The embodied haptic model leverages low-level haptic signals obtained from the robot's manipulator to make action predictions based on the human poses and forces collected with

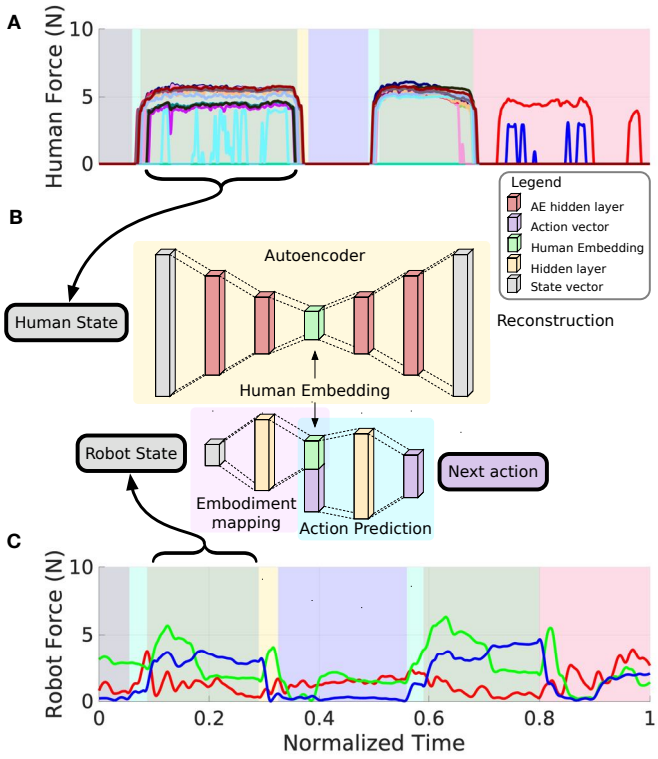


Figure 2.5: Illustration of learning embodied haptic representation and action prediction model. An example of the force information in (A) the human state, collected by the tactile glove (with 26 dimensions of force data), and force information in (C) the robot state, recorded from the force sensors in the robot's end-effector (with 3-dimensions of force data). The background colors indicate different action segments. (B) Embodied haptic representation and action prediction model. The autoencoder (yellow background) takes a human state, reduces its dimensionality to produce a human embedding, and uses the reconstruction to verify that the human embedding maintains the essential information of the human state. The embodiment mapping network (purple background) takes in a robot state and maps to an equivalent human embedding. The action prediction network (light blue background) takes the human embedding and the current action and predicts what action to take next. Copyright reserved to original publication [EGL19].

the tactile glove. This embodied haptic sensing allows the robot to reason about (i) its own haptic feedback by imagining itself as a human demonstrator, and (ii) what a human would have done under similar poses and forces. The critical challenge here is to learn a mapping between equivalent robot and human states, which is difficult due to the different embodiments. From the perspective of generalization, manually designed embodiment mappings are not desirable. To learn from human demonstrations on arbitrary robot embodiments, we propose an embodied haptic model general enough to learn between an arbitrary robot embodiment and a human demonstrator.

The embodied haptic model consists of three major components: (i) an autoencoder to encode the human demonstration in a low-dimensional subspace; we refer to the reduced embedding as the *human embedding*. (ii) An *embodiment mapping* that maps robot states onto a corresponding human embedding, providing the robot with the ability to imagine itself as a human demonstrator. (iii) An *action predictor* that takes a human embedding and the current action executing as the input and predicts the next action to execute, trained using the action labels from human demonstrations. Fig. 2.5B shows the embodied haptic network architecture. Using this network architecture, the robot infers what action a human was likely to execute based on this inferred human state. This embodied action prediction model picks the next action according to:

$$a_{t+1} \sim p(\cdot|f_t, a_t), \tag{2.1}$$

where a_{t+1} is the next action, f_t is the robot's current haptic sensing, and a_t is the current action.

The autoencoder network takes an 80-dimensional vector from the human demonstration (26 for the force sensors and 54 for the poses of each link in the human hand) and uses the post-condition vector, *i.e.*, the average of last N frames (we choose $N = 2$ to minimize the variance), of each action in the demonstration as input; see the Autoencoder portion of Fig. 2.5B. This input is then reduced to an 8-dimensional human embedding. Given a human

demonstration, the autoencoder enables the dimensionality reduction to an 8-dimensional representation.

The embodiment mapping maps from the robot's 4-dimensional post-condition vector, *i.e.*, the average of the last N frames (different from human post-condition due to a faster sample rate on the robot gripper compared to the tactile glove; we choose $N = 10$), to an imagined human embedding; see the Embodiment Mapping portion of Fig. 2.5B. This mapping allows the robot to imagine its current haptic state as an equivalent low-dimensional human embedding. The robot's 4-dimensional post-condition vector consists of the gripper position (1 dimension) and the forces applied by the gripper (3 dimensions). The embodiment mapping network uses a 256-dimensional latent representation, and this latent representation is then mapped to the 8-dimensional human embedding.

To train the embodiment mapping network, the robot first executes a series of supervised actions where if the action produces the correct final state of the action, the robot post-condition vector is saved as input for network training. Next, human demonstrations of equivalent actions are fed through the autoencoder to produce a set of human embeddings. These human embeddings are considered as the ground-truth target outputs for the embodiment mapping network, regardless of the current reconstruction accuracy of the autoencoder network. Then the robot execution data is fed into the embodiment mapping network, producing an imagined human embodiment. The embodiment mapping network optimizes to reduce the loss between its output from the robot post-condition input and the target output.

For the action predictor, the 8-dimensional human embedding and the 10-dimensional current action are mapped to a 128-dimensional latent representation, and the latent representation is then mapped to a final 10-dimensional action probability vector (*i.e.*, the next action); see Action Prediction portion of Fig. 2.5B. This network is trained using human demonstration data, where a demonstration is fed through the autoencoder to produce a human embedding, and that human embedding and the one-hot vector of the current action

execution are fed as the input to the prediction network; the ground-truth is the next action executed in the human demonstration.

The network in Fig. 2.5B is trained in an end-to-end fashion with three different loss functions in a two-step process: (i) a forward pass through the autoencoder to update the human embedding z_h. After computing the error $L_{\text{reconstruct}}$ between the reconstruction s'_h and the ground-truth human data s_h, we back-propagate the gradient and optimize the autoencoder:

$$L_{\text{reconstruct}}(s'_h, s_h) = \frac{1}{2}(s'_h - s_h)^2. \tag{2.2}$$

(ii) A forward pass through the embodiment mapping and the action prediction network. The embodiment mapping is trained by minimizing the difference L_{mapping} between the embodied robot embedding z_r and target human embedding z_h; the target human embedding z_h is acquired through a forward pass through the autoencoder using a human demonstration post-condition of the same action label, s_h. We compute the cross-entropy loss $L_{\text{prediction}}$ of the predicted action label a' and the ground-truth action label a to optimize this forward pass:

$$\begin{aligned} L_{\text{planning}}(a', a) &= L_{\text{mapping}} + \beta \cdot L_{\text{prediction}} \\ L_{\text{mapping}} &= \frac{1}{2}(z_r - z_h)^2 \\ L_{\text{prediction}} &= H(p(a'), q(a)), \end{aligned} \tag{2.3}$$

where H is the cross-entropy, p is the model prediction distribution, q is the ground-truth distribution, and β is the balancing parameter between the two losses.

2.2.1.1 Training Details of Embodied Haptic Model

In this section, we present the implementation detail for reproducibility.

Network Architecture The autoencoder is constructed with a multi-layer perceptron (MLP); see Table 2.1. The human embedding can be obtained with a forward pass through

the network. The supervision for the autoencoder is the original human post-condition. The loss is measured by the reconstruction error. The robot-human embodiment mapping is implemented with an MLP; see Table 2.2. The embodiment mapping is trained using equivalent human and robot post-conditions (equivalent here means the post-condition of executing the same action successfully). The human post-condition is fed through the autoencoder to produce a human embedding, and this embedding serves as the supervision target for the embodiment mapping network. The last major component of the embodied haptic prediction model is the action predictor, also implemented with an MLP; see Table 2.3. The supervision for the action predictor is the ground-truth human action labels.

Training Details We adopt a two-step updating schema for the embodied haptic model. In the first step, we feed forward the human post-condition data into the autoencoder. The encoder will reduce the high-dimensional human data to a low-dimensional human embedding; the encoder and the decoder are learned with hyper-parameter shown in Table 2.4. The supervision for the autoencoder is the reconstructed original human post-condition. In the second step, with the human embedding and the action labels, the action predictor and the embodiment mapping are training jointly with the hyper-parameters shown in Table 2.4. The embodiment mapping is trained using equivalent human and robot post-conditions (equivalent here means the post-condition of executing the same action successfully). The human post-condition is fed through the autoencoder to produce a human embedding, and this embedding serves as the supervision target for the embodiment mapping network. The supervision for the action predictor is the ground-truth human action labels.

A similar embodied haptic model was presented in [EGX17] but with two separate loss functions, which is more difficult to train compared to the single loss function presented in this chapter. A clear limitation of the haptic model is the lack of long-term action planning. To address this problem, we discuss the symbolic task planner below and then discuss how we integrate the haptic model with the symbolic planner to jointly find the optimal action.

Table 2.1: Network architecture and parameters of the autoencoder. Network architecture is defined from the top of the table to the bottom, with the first and last layer being input and output, respectively.

Operator	**Params**
Linear	80
ReLU	
Linear	64
ReLU	
Linear	16
ReLU	
Linear	8
ReLU	
Linear	16
ReLU	
Linear	64
ReLU	
Linear	80

2.2.2 Symbolic Action Planner

Opening medicine bottles is a challenging multi-step manipulation, as one may need to push on the cap to unlock it (visually unobservable), twist it, and then pull it open. A symbolic representation is advantageous to capture the necessary long-term constraints of the task. From labeled action sequences of human demonstrations, we induce a T-AOG, a probabilistic graphical model describing a stochastic, hierarchical, and compositional context-free grammar [ZM07], wherein an And-node encodes a decomposition of the graph into sub-graphs, an Or-node reflects a switch among multiple alternate sub-configurations, and the terminal

Table 2.2: Network architecture and parameters for robot to human embedding. Network architecture is defined from the top of the table to the bottom, with the first and last layer being input and output, respectively.

Operator	**Params**
Linear, Linear	3, 1
ReLU, ReLU	
Linear, Linear	128, 128
ReLU	
Linear	8

Table 2.3: Network architecture and parameters for action prediction. Network architecture is defined from the top of the table to the bottom, with the first and last layer being input and output, respectively.

Operator	**Params**
Linear, Linear	8, 13
ReLU, ReLU	
Linear, Linear	64, 64
ReLU	
Linear	10

nodes consist of a set of action primitives (such as *push*, *twist*, *pull*). A corpus of sentences (*i.e.*, action sequences in our case) is fed to the grammar induction algorithm presented in [TPZ13], and the grammar is induced by greedily generating And-Or fragments according to the data likelihood; the fragments represent compositional substructures that are combined to form a complete grammar. In our case, the grammar is learned from segmented and labeled human demonstrations. The resulting grammar offers a compact symbolic rep-

Table 2.4: Hyper-parameters used during training.

Parameter	**Value**
Autoencoder learning rate	5e-5
Action predictor learning rate	5e-5
Balance param. (β)	1
Batch size	16
No. of epochs	150

resentation of the task and captures the hierarchical structure of the task, including different action sequences for different bottles, as well as different action sequences for the same bottle. Examples of the T-AOG learning progress are shown in Fig. 2.6. The nodes connected by red edges in Fig. 2.6C indicate a parse graph sampled from the grammar, and its terminal nodes compose an action sequence for robot execution.

Based on the action sequences observed in human demonstrations, the induced grammar can be used to parse and predict robot action sequences likely to lead to opening the medicine bottle successfully, assuming each robot action corresponds to an equivalent human action. The induced grammar can be parsed to generate new, unseen, and valid action sequences for solving similar tasks (*e.g.*, opening different medicine bottles), and thus the grammar can be used with symbolic planning methods, such as the Earley Parser [QJZ18]. We denote the process of planning actions using a parser and the action grammar as the *symbolic planner*. Hence, the symbolic planner endows the robot with the ability to ask itself from a *mechanistic* perspective: *based on what I have done thus far and what I observed the human do, which actions are likely to open the bottle at the end of the sequence?*

The symbolic planner utilizes stochastic context-free grammars to represent tasks, where the terminal nodes (words) are actions and sentences are action sequences. Given an action grammar, the planner finds the optimal action to execute next based on the action history,

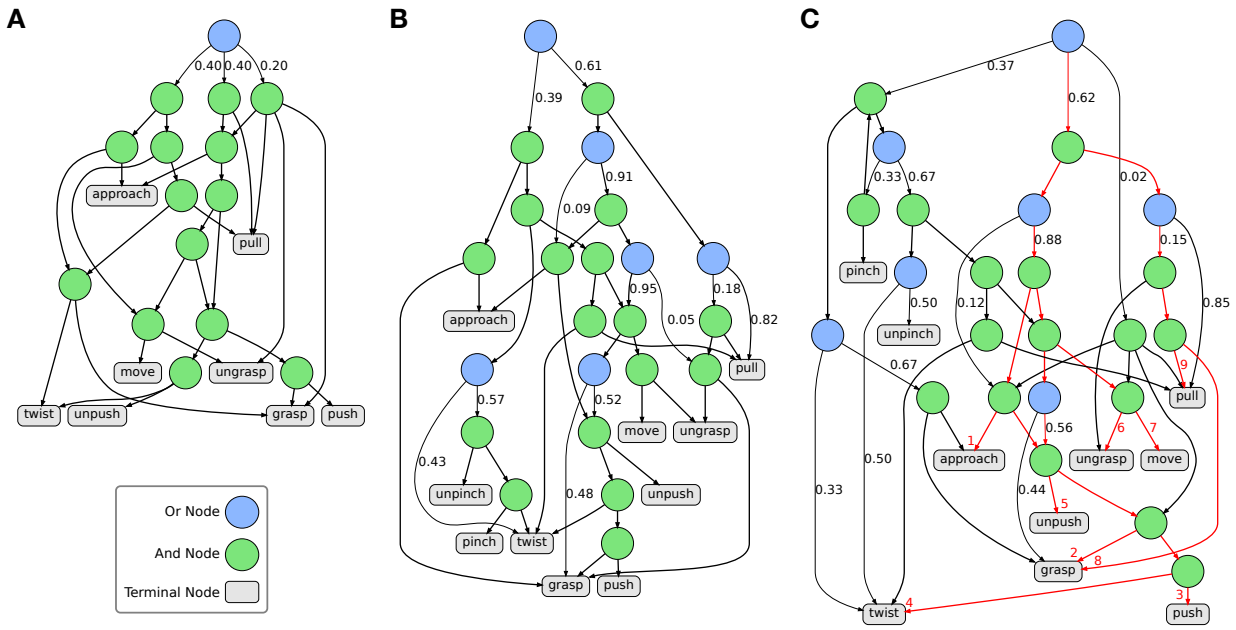


Figure 2.6: An example of action grammar induced from human demonstrations. Green nodes represent And-nodes, and blue nodes represent Or-nodes. Probabilities along edges emanating from Or-nodes indicate the parsing probabilities of taking each branch. Grammar model induced from (A) 5 demonstrations, (B) 36 demonstrations, (C) 64 demonstrations. The grammar model in (C) also shows a parse graph highlighted in red, where red numbers indicate temporal ordering of actions. Copyright reserved to original publication [EGL19].

analogous to predicting the next word given a partial sentence.

The action grammar is induced using labeled human demonstrations, and we assume the robot has an equivalent action for each human action. Each demonstration forms a sentence, x_i, and the collection of sentences from a corpus, $x_i \in X$. The segmented demonstrations are used to induce a stochastic context-free grammar using the method presented in [TPZ13]. This method pursues T-AOG fragments to maximize the likelihood of the grammar producing the given corpus. The objective function is the posterior probability of the grammar given

the training data X:

$$p(G|X) \propto p(G)p(X|G) = \frac{1}{Z}e^{-\alpha||G||} \prod_{x_i \in X} p(x_i|G), \tag{2.4}$$

where G is the grammar, $x_i = (a_1, a_2, \ldots, a_m) \in X$ represents a valid sequence of actions with length m from the demonstrator, α is a constant, $||G||$ is the size of the grammar, and Z is the normalizing factor. Fig. 2.6 shows examples of induced grammars of actions.

During the symbolic planning process, this grammar is used to compute which action is the most likely to open the bottle based on the action sequence executed thus far and the space of possible future actions. A pure symbolic planner picks the optimal action based on the grammar prior:

$$a^*_{t+1} = \arg\max_{a_{t+1}} p(a_{t+1} \,|\, a_{0:t}, G), \tag{2.5}$$

where a_{t+1} is the next action, and $a_{0:t}$ is the action sequence executed thus far. This grammar prior can be obtained by a division of two grammar prefix probabilities: $p(a_{t+1} \,|\, a_{0:t}, G) = \frac{p(a_{0:t+1} \,|\, G)}{p(a_{0:t} \,|\, G)}$, where the grammar prefix probability $p(a_{0:t} \,|\, G)$ measures the probability that $a_{0:t}$ occurs as a prefix of an action sequence generated by the action grammar G. Based on a classic parsing algorithm—the Earley parser [Ear70]—and dynamic programming, the grammar prefix probability can be obtained efficiently by the Earley-Stolcke parsing algorithm [Sto95]. An example of pure symbolic planning is shown in Fig. 2.7.

However, due to the fixed structure and probabilities encoded in the grammar, always choosing the action sequence with the highest grammar prior is problematic since it provides no flexibility. An alternative pure symbolic planner picks the next action to execute by sampling from the grammar prior:

$$a_{t+1} \sim p(\cdot \,|\, a_{0:t}, G). \tag{2.6}$$

In this way, the symbolic planner samples different grammatically correct action sequences and increases the adaptability of the symbolic planner. In the experiments, we choose to sample action sequences from the grammar prior.

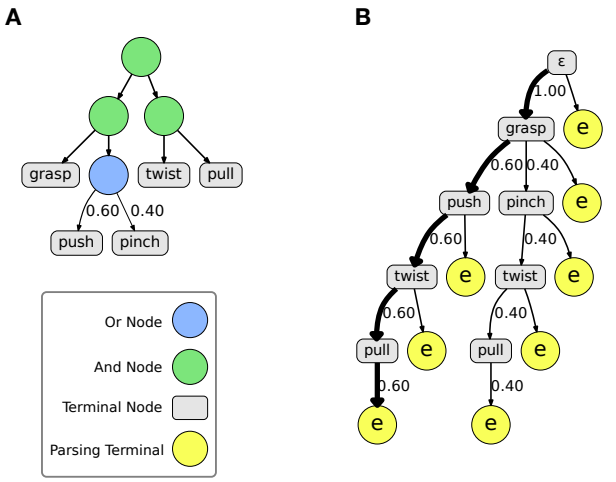


Figure 2.7: Action grammars and grammar prefix trees used for parsing. (A) An example action grammar. (B) A grammar prefix tree with grammar priors. The numbers along edges are the prefix or parsing probabilities of the action sequence represented by the path from the root node to the node pointed by the edge. When the corresponding child node of an edge is an action terminal, the number along the edge represents a prefix probability; when the corresponding child is a parsing terminal e, the number represents the parsing probability of the entire sentence. In this example, the action sequence "grasp, push, twist, pull" has the highest probability of 0.6. The root ϵ represents the empty symbol where no terminals were parsed. Copyright reserved to original publication [EGL19].

In contrast to the haptic model, this symbolic planner lacks the adaptability to real-time sensor data. However, this planner encodes long-term temporal constraints that are missing from the haptic model, since only grammatically correct sentences have non-zero probabilities. The GEP adopted in this chapter naturally combines the benefits of both the haptic model and the symbolic planner; see the next section.

2.2.3 Integration of Symbolic Planner and Haptic Model

To integrate the long-term task structure induced by the symbolic planner and manipulation strategy learned from haptic signals, we seek to combine the symbolic action planner and

embodied haptic model using the generalized Earley parser (GEP) [QJZ18]. The GEP is a grammar parser that works on a sequence of sensory data; it combines any context-free grammar model with probabilistic beliefs over possible labels (grammar terminals) of sensory data. The output of the GEP is the optimal segmentation and label sentence of the raw sensory data; a label sentence is optimal when its probability is maximized according to the grammar priors and the input belief over labels while being grammatically correct. The core idea of the GEP is to efficiently search in the language space defined by the grammar to find the optimal label sentence.

To adopt the GEP for a robot system, we modify the GEP presented in [QJZ18] for online planning. The grammar for the GEP remains the same grammar used in the symbolic planner; however, the GEP's probabilistic beliefs come from the softmax distribution from the haptic model. During the action planning process, a stochastic distribution of action labels predicted by the haptic model is fed into the GEP at every time step. The GEP aggregates the entire symbolic planning history with the current haptic prediction and outputs the best parse to plan the most likely next action. Intuitively, such an integration of the symbolic planner and haptic model enables the robot to ask itself: *based on the human demonstration, the poses and forces I perceive right now and the action sequence I have executed thus far, which action has the highest likelihood of opening the bottle?*

The integrated GEP model finds the next optimal action considering both the action grammar G and the haptic input f_t:

$$a^*_{t+1} = \arg\max_{a_{t+1}} p(a_{t+1} \mid a_{0:t}, f_t, G). \tag{2.7}$$

Conceptually, this can be thought of as a posterior probability that considers both the grammar prior and the haptic signal likelihood. The next optimal action is computed by an improved generalized Earley parser (GEP) [QJZ18]; GEP is an extension of the classic Earley parser [Ear70]. In the present work, we further extend the original GEP to make it applicable to multisensory inputs and provide an explanation in real-time for robot systems, instead of for offline video processing.

The computational process of GEP is to find the optimal label sentence according to both a grammar and a classifier output of probabilities of labels for each time step. In our case, the labels are actions, and the classifier output is given by the haptic model. Optimality here means maximizing the joint probability of the action sequence according to the grammar prior and haptic model output while being grammatically correct.

The core idea of the algorithm is to directly and efficiently search for the optimal label sentence in the language defined by the grammar. The grammar constrains the search space to ensure that the sentence is always grammatically correct. Specifically, a heuristic search is performed on the prefix tree expanded according to the grammar, where the path from the root to a node represents a partial sentence (prefix of an action sequence).

GEP is a grammar parser, capable of combining the symbolic planner with low-level sensory input (haptic signals in this chapter). The search process in the GEP starts from the root node of the prefix tree, which is an empty terminal symbol indicating no terminals are parsed. The search terminates when it reaches a leaf node. In the prefix tree, all leaf nodes are parsing terminals e that represent the end of parse, and all non-leaf nodes represent terminal symbols (*i.e.*, actions). The probability of expanding a non-leaf node is the prefix probability, *i.e.*, how likely is the current path being the prefix of the action sequence. The probability of reaching a leaf node (parsing terminal e) is the parsing probability, *i.e.*, how likely is the current path to the last non-leaf node being the executed actions and the next action. In other words, the parsing probability measures the probability that the last non-leaf node in the path will be the next action to execute. It is important to note that this prefix probability is computed based on both the grammar prior and the haptic prediction; in contrast, in the pure symbolic planner, the prefix probability is computed based on only the grammar prior. An example of the computed prefix and parsing probabilities and output of GEP is given by Fig. 2.8. For an algorithmic description of this process, see [EGL19].

The original GEP is designed for offline video processing. In this chapter, we made modifications to enable online planning for a robotic task. The major difference between parsing

A Input probability matrix

Time step	*grasp*	*push*	*pinch*	*twist*	*pull*
0	0.6	0.1	0.1	0.1	0.1
1	0.6	0.1	0.1	0.1	0.1
2	0.1	0.1	0.6	0.1	0.1
3	0.1	0.1	0.6	0.1	0.1
4	0.1	0.1	0.1	0.6	0.1
5	0.1	0.1	0.1	0.6	0.1

B Cached probabilities

Time step	ϵ	*grasp*	*grasp push*	*grasp pinch*	*grasp push twist*	*grasp pinch twist*	*grasp push twist pull*	*grasp pinch twist pull*
0	0.000	0.600	0.000	0.000	0.000	0.000	0.000	0.000
1	0.000	0.360	0.036	0.024	0.000	0.000	0.000	0.000
2	0.000	0.036	0.025	0.101	0.004	0.002	0.000	0.000
3	0.000	0.004	0.005	0.069	0.003	0.010	3.6e-04	2.4e-04
4	0.000	3.6e-04	6.8e-04	0.007	0.005	0.048	3.2e-04	0.001
5	0.000	3.6e-05	9.0e-05	7.2e-04	0.003	0.033	4.9e-04	0.005
prefix	1.000	0.600	0.060	0.119	0.009	0.058	0.001	0.006

C

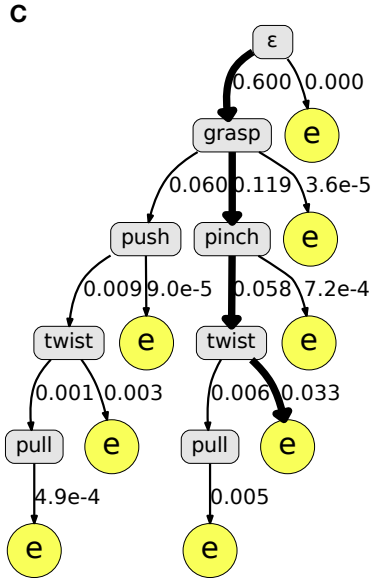


Figure 2.8: An example of the generalized Earley parser (GEP). (A) A classifier is applied to a 6-frame signal and outputs a probability matrix as the input. (B) A table of the cached probabilities of the algorithm. For all expanded action sequences, it records the parsing probabilities at each time step and prefix probabilities. (C) Grammar prefix tree with the classifier likelihood. The GEP expands a grammar prefix tree and searches in this tree. It finds the best action sequence when it hits the parsing terminal e. It finally outputs the best label "grasp, pinch, pull" with a probability 0.033. The probabilities of children nodes do not sum to 1 because grammatically incorrect nodes are eliminated from the search and the probabilities are not re-normalized [QJZ18]. Copyright reserved to original publication [EGL19].

and planning is the uncertainty about past actions: there is uncertainty about observed actions during parsing. However, during planning, there is no uncertainty about executed actions—the robot directly chooses which actions to execute, thereby removing any ambiguity regarding which action was executed at a previous timestep. Hence, we need to prune the impossible parsing results after executing each action; each time after executing an action, we change the probability vector of that action to a one-hot vector. This modification

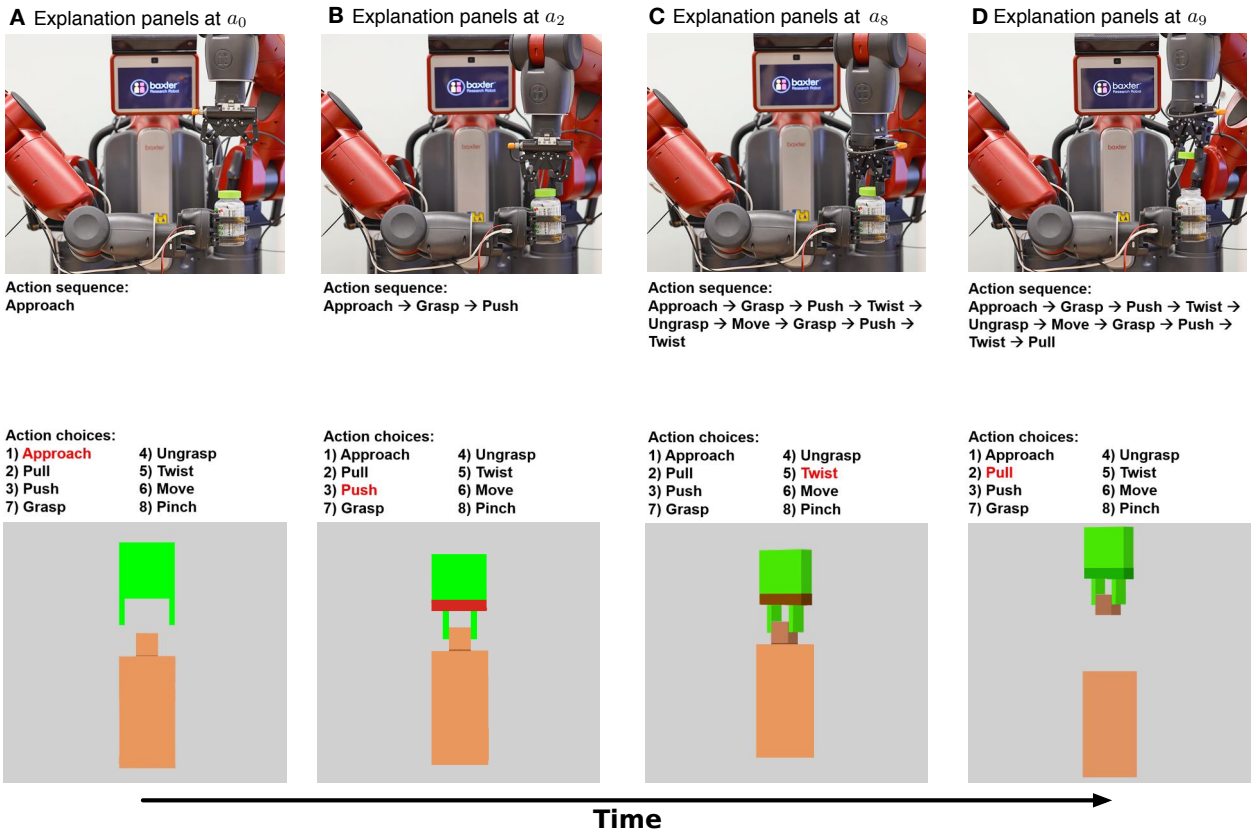


Figure 2.9: Explanations generated by the symbolic planner and the haptic model. (A) Symbolic (mechanistic) and haptic (functional) explanations at a_0 of the robot action sequence. (B), (C), and (D) show the explanations at times a_2, a_8, and a_9, where a_i refers to the ith action. Note that the red on the robot gripper's palm indicates a large magnitude of force applied by the gripper, and green indicates no force; other values are interpolated. These explanations are provided in real-time as the robot executes. Copyright reserved to original publication [EGL19].

effectively prunes the action sequences that contain the impossible actions executed thus far by the robot.

2.3 Explanation Generation

The haptic model and symbolic planner are capable of providing explanations to humans about robot behavior in real-time. Mechanistic explanations can be generated by the symbolic planner in the form of action sequences as they represent the process of opening a medicine bottle. Functional explanations can be provided by a visualization of the internal robot gripper state (effects) used in the haptic model. It is worth noting that these models are capable of *providing* such explanations but are not the only means of producing them. Alternative action planners and haptic models could produce similar explanations, as long as the robot systems are able to learn the corresponding representations for haptic prediction and task structure. Fig. 2.9 shows the explanation panels over an action sequence. These visualizations are shown in real-time, providing direct temporal links between explanation and execution.

To visualize the forces imposed by the robot gripper, we first identify the max force magnitude in all the force signal data collected from human demonstrations. Then, all force data is normalized to the value between 0 and 1, where 0 corresponds to pure green in the visualization, and 1 pure red. The value in between is interpolated linearly and displayed on the robot's palm.

2.4 Results

2.4.1 Robot Results

Fig. 2.10A and Fig. 2.10B show the success rate of the robot in performing the task of opening the 3 medicine bottles used in human demonstrations and 2 new, unseen medicine bottles, respectively; see more generalization results in Fig. 2.11. The 2 generalization bottles locking mechanisms that are similar (but not identical) to the ones used in human demonstrations, and the low-level haptic signals are significantly different, posing challenges in transferring

the learned knowledge to novel unseen cases. Each bottle and model was executed 31 times on our robot platform. In the testing experiments, Bottle 1 is a regular bottle without a locking mechanism, Bottle 2 has a *push-twist* locking mechanism, and Bottle 3 requires pinching *specific points* on the lid to unlock. In the generalization experiments, Bottle 4 also does not have a locking mechanism, and Bottle 5 has a *push-twist* locking mechanism but with a different shape, size, and haptic signals compared with the ones in the human demonstrations. For both the testing and generalization experiments, the robot's task performance measured by the success rates decreases as the bottle's locking mechanism becomes more complex, as expected.

To quantitatively compare the difference between the model components, we conduct ablative experiments on robot task performance using only the symbolic planner and only the haptic model; see Fig. 2.10. The haptic model and symbolic planner vary in their relative individual performance, but the combined planner using the GEP yields the best performance for all cases. Hence, integrating both the long-term task structure provided by the symbolic planner and the real-time sensory information provided by the haptic model yields the best robot performance. The symbolic planner provides long-term action planning and ensures the robot executes an action sequence capturing the high-level structure of the task. However, models that solely rely on these symbolic structures are brittle to adjust to perturbations of haptic signals, especially when the task relies more on the haptics as the complexity increases. On the other hand, models that rely purely on haptic signals are unable to impose multi-step task constraints, and thus may fail to infer a correct sequence of actions based on the execution history. Our results confirm that by combining these modalities together, the robot achieves the highest task performance.

Given that multiple modalities are involved in the GEP's performance, it is crucial to assess the contributions from different model components. We ran the χ^2-test to determine if different models (GEP, symbolic, and haptic) are statistically different in their ability to open five bottles (3 bottles used in human demonstrations and 2 new bottles used in

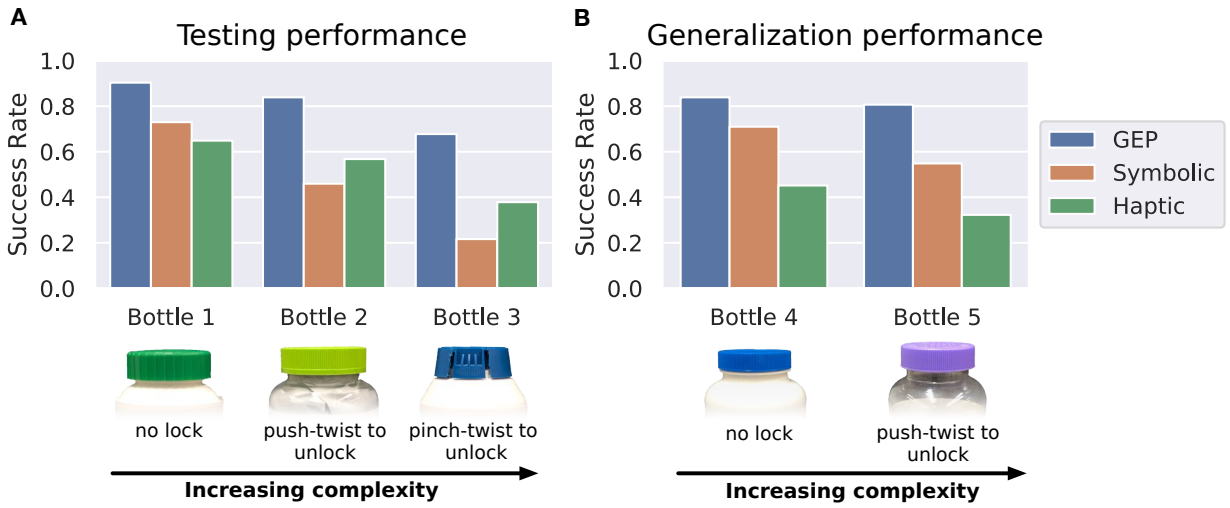


Figure 2.10: Robot task performance on different bottles with various locking mechanisms using the symbolic planner, haptic model, and the GEP that integrates both. (A) Testing performance on bottles observed in human demonstrations. Bottle 1 does not have a locking mechanism, Bottle 2 employs a *push-twist* locking mechanism, and Bottle 3 employs a *pinch-twist* locking mechanism. (B) Generalization performance on new, unseen bottles. Bottle 4 does not have a locking mechanism, and Bottle 5 employs a *push-twist* locking mechanism. The bottles used in generalization have similar locking mechanisms but evoke significantly different haptic feedback. Regardless of testing on demonstration or unseen bottles, the best performance is achieved by the GEP that combines the symbolic planner and haptic model. Copyright reserved to original publication [EGL19].

the generalization task). The robot performs the manipulation task 31 times per medicine bottle. With the significance level of 0.05, the results show that the performance of the GEP model is significantly better than both symbolic model ($\chi^2(1) = 10.0916, p = 0.0015$) and haptic model ($\chi^2(1) = 13.0106, p < 0.001$). Performance does not show difference between the symbolic model and the haptic model, $\chi^2(1) = 0.1263, p = 0.7232$. These results suggest that both haptic model and symbolic planner contribute to good task performance; when the two processes are integrated with the GEP, the success rate of the robot for opening

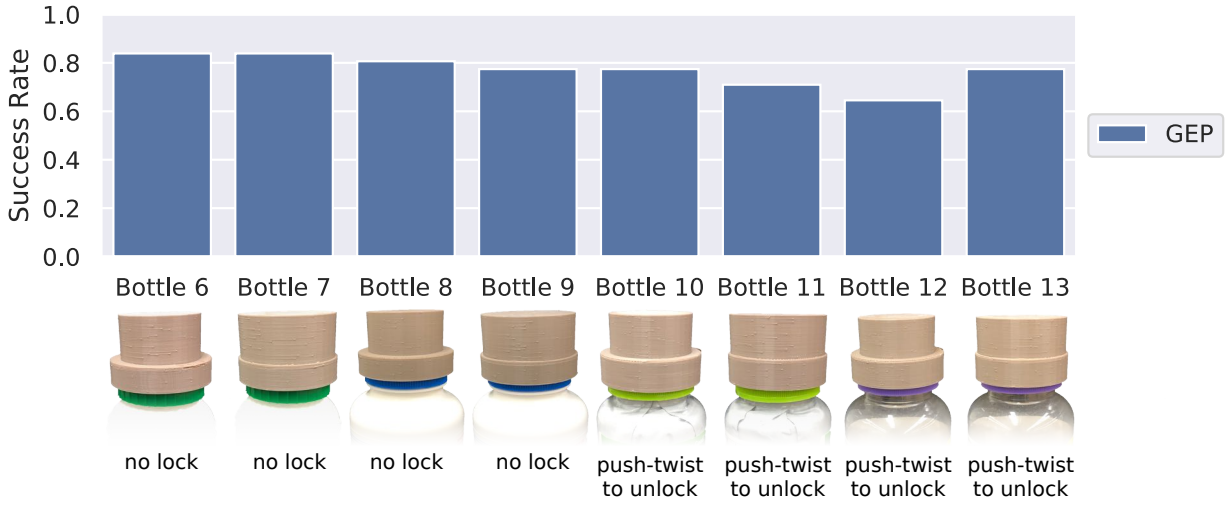


Figure 2.11: Additional generalization experiments on bottles augmented with different 3D-printed caps. The GEP shows good performance across all bottles, indicating the GEP is able to generalize to bottles with similar locking mechanisms as in the human demonstrations, but significantly different haptic signals. Copyright reserved to original publication [EGL19].

medicine bottles is improved compared to the performance by the single-module models based on either the haptic model or the symbolic planner.

2.4.2 Human Experiment

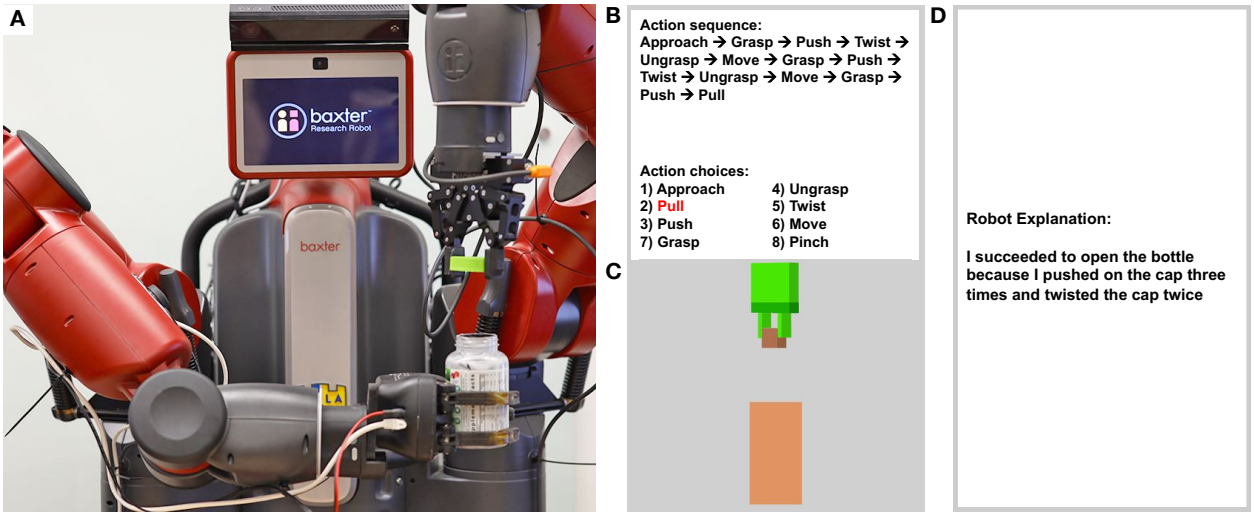
2.4.2.1 Experimental Design

The human experiment aims to examine whether providing explanations generated from the robot's internal decisions fosters human trust in machines and what forms of explanation are the most effective in enhancing human trust. We conducted a psychological study with 150 participants; each was randomly assigned to one of five groups. Our experimental setup consisted of two phases: familiarization and prediction. During familiarization, all groups viewed the RGB video, and some groups were also provided with an explanation panel. During the second phase of the prediction task, all groups only observed RGB videos.

The five groups consist of the baseline no-explanation group, symbolic explanation group, haptic explanation group, GEP explanation group, and text explanation group. For the baseline no-explanation group, participants only viewed RGB videos recorded from a robot attempting to open a medicine bottle, as shown in Fig. 2.12A. For the other four groups, participants viewed the same RGB video of robot executions and simultaneously were presented with different explanatory panels on the right side of the screen. Specifically, the symbolic group viewed the symbolic action planner illustrating the robot's real-time inner decision-making, as shown in Fig. 2.12B. The haptic group viewed the real-time haptic visualization panel, as shown in Fig. 2.12C. The GEP group viewed the combined explanatory panel, including the real-time robot's symbolic planning and an illustration of haptic signals from the robot's manipulator, namely both Fig. 2.12B-C. The text explanation group was provided a text description that summarizes why the robot succeeded or failed to open the medicine bottle *at the end* of the video, as shown in Fig. 2.12D. See a summary in Fig. 2.12E for the five experimental groups.

During the familiarization phase, participants were provided two demonstrations of robot executions, with one successful execution of opening a medicine bottle and one failed execution without opening the same bottle. The presentation order of the two demonstrations was counterbalanced across participants. After observing robot executions with explanation panels, participants were first asked to provide a trust rating for the question: *to what extent do you* ***trust/believe*** *this robot possesses the ability to open a medicine bottle?* on a scale between 0 and 100. The question was adopted from the questionnaire of measuring human trust in automated systems [JBD00]. This question also clearly included the goal of the system, *to open a medicine bottle*, to enhance the reliability in trust measures [CF98]. Hence, the rating provided a direct qualitative measure of human trust in the robot's ability to open medicine bottles.

In addition, we designed the second measure to assess the quantitative aspects of trust. We adopted the definition by Castelfranchi and Falcone [CF98] that quantitative trust is



E Summary of human subject groups and explanations presented

Group	RGB (A)	Symbolic (B)	Haptic (C)	Text (D)
Baseline	✓			
Symbolic	✓	✓		
Haptic	✓		✓	
GEP	✓	✓	✓	
Text	✓			✓

Figure 2.12: Illustration of visual stimuli used in human experiment. All five groups observed the RGB video recorded from robot executions, but differed by the access to various explanation panels. (A) RGB video recorded from robot executions. (B) Symbolic explanation panel. (C) Haptic explanation panel. (D) Text explanation panel. (E) A summary of which explanation panels were presented to each group. Copyright reserved to original publication [EGL19].

based on the quantitative dimensions of its cognitive constituents. Specifically, the greater the human's belief in the machine's competence and performance, the greater the human trust in machines. In the prediction phase, we asked participants to predict the robot's next action in a new execution with the same task of opening the same medicine bottle. Participants viewed different segments of actions performed by the robot and were asked to answer the prediction question over time. For this measure, participants in all five groups only observed RGB videos of robot execution during the prediction phase; no group had

access to any explanatory panel after the familiarization phase. The prediction accuracy was computed as the quantitative measure of trust, with the presumption that, as the robot behavior is more predictable to humans, greater prediction accuracy indicates higher degrees of trust.

Human participants were recruited from the University of California, Los Angeles (UCLA) Department of Psychology subject pool and were compensated with course credit for their participation. A total of 163 students were recruited, each randomly assigned to one of the five experimental groups. Thirteen participants were removed from the analysis for failing to understand the haptic display panel by not passing a recognition task. Hence, the analysis included 150 participants (mean age of 20.7). The symbolic and haptic explanation panels were generated as described in Section 2.3. The text explanation was generated by the authors based on the robot's action plan to provide an alternate text summary of robot behavior. Although such text descriptions were not directed yielded by the model, they could be generated by modern natural language processing methods.

The human experiment included two phases: familiarization and prediction. In the familiarization phase, participants viewed two videos showing a robot interacting with a medicine bottle, with one successful attempt of opening the bottle and one failed attempt without opening the bottle. In addition to the RGB videos showing the robot's executions, different groups viewed the different forms of explanation panels. At the end of familiarization, participants were asked to assess how well they trusted/believed the robot possessed the ability to open the medicine bottle; see Fig. 2.13 for the illustration of the trust rating question.

Next, the prediction phase presented all groups with only RGB videos of a successful robot execution; no group had access to any explanatory panels. Specifically, participants viewed videos segmented by the robot's actions; for segment i, videos start from the beginning of the robot execution up to the ith action. For each segment, subjects were asked to predict what action the robot would execute next; see Fig. 2.14 for an illustration of the action prediction question.

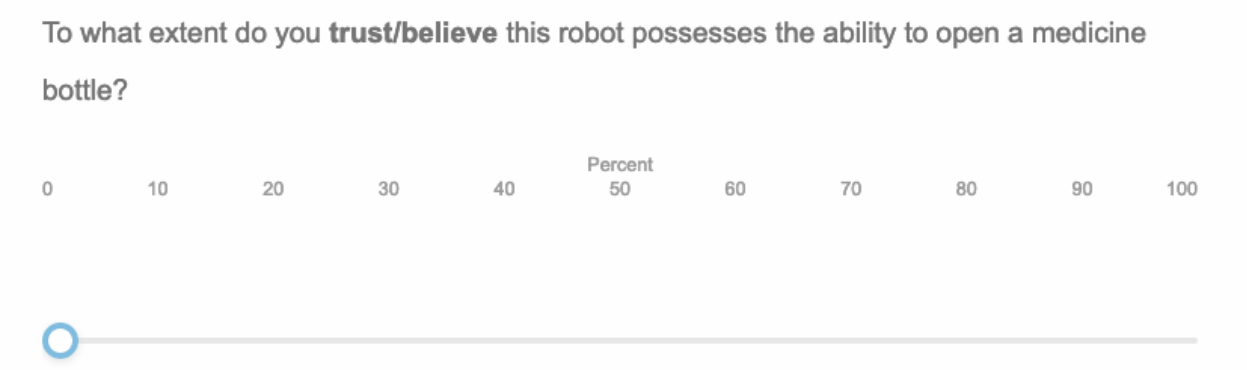


Figure 2.13: Qualitative trust question asked to human subjects after observing two demonstrations of robot execution. This question was immediately asked after the familiarization phase of the experiment; in other words, we asked this question immediately after the subjects had observed robot executions *with* access to the explanation panel (if the subject's group had access to an explanation panel; *i.e.* all groups except baseline). Copyright reserved to original publication [EGL19].

The prediction phase evaluates how well each explanation panel imparts prediction ability after observing a robot's behaviors in solving the problem of opening a medicine bottle. Note that during the familiarization phase, the robot explains its behavior through explanatory panels, but during the prediction phase, subjects observe the robot executing the task with only the RGB videos. Thus our prediction question asks "*after familiarizing with explanatory panels, how well are human subjects able to predict robot behavior when observing only RGB robot executions?*" The prediction accuracy is computed as the percentage of correct action predictions in the sequence. This experimental design examines how well each explanatory panel imparts prediction ability under new robot executions where no explanation panel is available. For each question, participants selected from 8 actions: *push on the cap*, *pinch the cap*, *pull the cap*, *twist the cap*, *grasp the cap*, *ungrasp the cap*, *move the left robot arm to grasping position*, and *nothing*.

Regardless of group assignment, all RGB videos were the same across all groups; *i.e.*, we show the same RGB video for all groups with varying explanation panels. This experimental

Figure 2.14: Prediction accuracy question asked to human subjects after each segment of the robot's action sequence during the prediction phase of the experiment. No group had access to explanation panels during the prediction phase; subjects had to predict the action while only observing RBG videos of each action segment. Copyright reserved to original publication [EGL19].

design isolates potential effects of execution variations in different robot execution models presented in Section 2.2; we only seek to evaluate how well explanation panels foster qualitative trust and enhance prediction accuracy and keep robot execution performance constant across groups to remove potential confounding.

For both qualitative trust and prediction accuracy, the null hypothesis is that the explanation panels foster equivalent levels of trust and yield the same prediction accuracy across different groups, and therefore no difference in trust or prediction accuracy would

be observed. The test is a two-tailed independent samples t-test to compare performance from two groups of participants, as we used between-subjects design in the study, with a commonly used significance level $\alpha = 0.05$, assuming t-distribution, and the rejection region is $p < 0.05$.

2.4.2.2 Human Study Results

Fig. 2.15A shows human trust ratings from the five different groups. The analysis of variance (ANOVA) reveals a significant main effect of groups ($F(4, 145) = 2.848; p = 0.026$) with the significance level of 0.05. This result suggests that providing explanations about robot behavior in different forms impacts the degree of human trust in the robot system. Furthermore, we find that the GEP group with both symbolic and haptic explanation panels yields the highest trust rating, with a significantly better rating than the baseline group in which explanations are not provided (independent-samples t-test, $t(58) = 2.421; p = 0.019$). Interestingly, the GEP group shows a greater trust rating than the text group in which a summary description is provided to explain the robot behavior ($t(58) = 2.352; p = 0.022$), indicating detailed explanations of the robot's internal decisions over time is much more effective in fostering human trust than a summary text description to explain robot behavior. In addition, trust ratings in the symbolic group are also higher than ratings in the baseline group ($t(58) = 2.269; p = 0.027$) and higher than ratings in the text explanation group ($t(58) = 2.222; p = 0.030$), suggesting symbolic explanations play an important role in fostering human trust of the robot system. However, the trust ratings in the haptic explanation group are not significantly different from the baseline group, implying that explanations based only on haptic signals are not effective ways to gain human trust despite the explanations are also provided in real-time. No other significant group differences are observed between any other pairing of the groups.

The second trust measure based on prediction accuracy yields similar results. All groups provide action predictions above the chance-level performance of 0.125 (as there are 8 actions

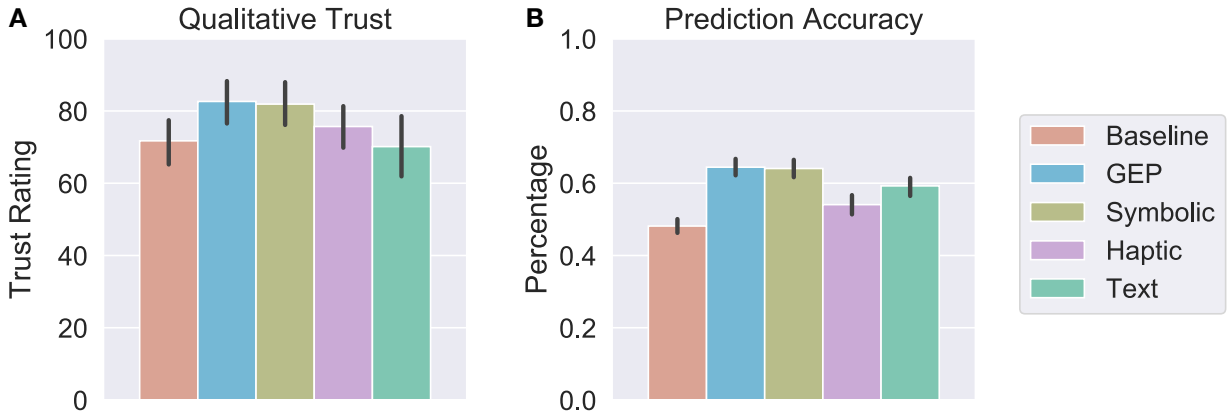


Figure 2.15: Human results for trust ratings and prediction accuracy. (A) Qualitative measures of trust: average trust ratings for the five groups. and (B) Average prediction accuracy for the five groups. The error bars indicate the 95% confidence interval. Across both measures, the GEP performs the best. For qualitative trust, the text group performs most similarly to the baseline group. For a tabular summary of the data, see [EGL19]. Copyright reserved to original publication [EGL19].

to choose from), showing that humans are able to predict the robot's behavior after only a couple of observations of a robot performing a task. The ANOVA analysis shows a significant main effect of groups ($F(4, 145) = 3.123; p = 0.017$), revealing the impact of provided explanations on the accuracy of predicting the robot's actions. As shown in Fig. 2.15B, participants in the GEP group yield significantly higher prediction accuracy than those in the baseline group ($t(58) = 3.285; p = 0.002$). Prediction accuracy of the symbolic group also yields better performance than the baseline group ($t(58) = 2.99; p = 0.004$). Interestingly, we find that the text group shows higher prediction accuracy than the baseline group ($t(58) = 2.144; p = 0.036$). This result is likely due to the summary text description providing a loose description of the robot's action plan; such a description decouples the explanation from the temporal execution of the robot. The prediction accuracy data did not reveal any other significant group differences among other pairs of groups.

In general, humans appear to need real-time, symbolic explanations of the robot's internal

decisions for performed action sequences in order to establish trust in machines when performing multi-step complex tasks. Summary text explanations and explanations only based on haptic signals are not effective ways to gain human trust, and the GEP and symbolic group foster similar degrees of human trust to the robot system according to both measures of trust.

2.5 Conclusion and Discussion

In terms of performance, our results demonstrate that a robot system can learn to solve challenging tasks from a small number of human demonstrations of opening three medicine bottles. This success in learning from small data is primarily supported by learning multiple models for joint inference of task structure and sensory predictions. We found that neither purely symbolic planning nor a haptic model is as successful as an integrated model including both processes.

Our model results also suggest that the relative contributions from individual modules, namely the symbolic planner and haptic predictions, can be influenced by the complexity of the manipulation task. For example, in testing scenarios, for Bottle 1 with no safety locking mechanism, the symbolic planner slightly outperforms the haptic model. Conversely, to open Bottle 3 that has complex locking mechanisms, the haptic model outperforms the symbolic planner as haptic signals provide critical information for the pinch action needed to unlock the safety cap. For generalization scenarios with new medicine bottles that are unseen in human demonstrations, the symbolic planner maintains a similar performance compared to equivalent bottles in the testing scenarios, whereas the haptic model performance decreases significantly. We also note that the symbolic planner performance decreases faster as complexity increases, indicating pure symbolic planners are more brittle to circumstances that require additional haptic sensing. Furthermore, as bottle complexity increases, model performance benefits more from integrating the symbolic planner and haptic signals. This trend

suggests that more complex tasks require the optimal combination of multiple models to produce effective action sequences.

In terms of explainability, we found that reasonable explanations generated by the robot system are important for fostering human trust in machines. Our experiments show that human users place more trust in a robot system that has the ability to provide explanations using symbolic planning. An intriguing finding from these experiments is that providing explanations in the form of a summarized text description of robot behavior is not an effective way to foster human trust. The symbolic explanation panel and text summary panel both provide critical descriptions of the robot's behavior at the abstract level, explaining why a robot succeeded or failed the task. However, the explanations provided by the two panels differ in their degree of detail and temporal presentation. The text explanation provides a loose description of the important actions that the robot executes after the robot finished the sequence. In contrast, the symbolic explanation included in the GEP's panel provides human participants with real-time internal decisions that the robot is planning to execute at each step. This mode of explanation enables the visualization of task structure for every action executed during the sequence and likely evokes a sense that the robot actively makes rational decisions.

However, it is not the case that a detailed explanation is always the best approach to foster human trust. A functional explanation of real-time haptic signals is not very effective in gaining human trust in this particular task. Information at the haptic level may be excessively tedious and may not yield a sense of rational agency that allows the robot to gain human trust. To establish human trust in machines and enable humans to predict robot behaviors, it appears that an effective explanation should provide a symbolic interpretation and maintain a tight temporal coupling between the explanation and the robot's immediate behavior.

Taking together both performance and explanation, we found that the relative contributions of different models for generating explanations may differ from their contributions to

maximizing robot performance. For task performance, the haptic model plays an important role for the robot to successfully open a medicine bottle with high complexity. However, the major contribution to gain human trust is made by real-time mechanistic explanations provided by the symbolic planner. Hence, model components that impart the most trust do not necessarily correspond to those components contributing to the best task performance. This divergence is intuitive as there is no requirement that components responsible for generating better explanations are the same components contributing to task performance; they are optimizing different goals. This divergence also implies that the robotics community should adopt model components that gain human trust, while also integrating these components with high-performance ones to maximize both human trust and successful execution. Robots endowed with explainable models offer an important step towards integrating robots into daily life and work.

CHAPTER 3

Generalization and Transfer in Causal Learning

In this chapter, we examine a learning problem in an interventional setting and show human learns are capable of causal abstraction, show model-free reinforcement is not, and present a hierarchical Bayesian capable of achieving near-human performance. The ability of agents to learn and *reuse* knowledge is a fundamental characteristic of general intelligence and is essential for agents to succeed in novel circumstances [LH07]. The key research question in the field of causal learning is how various intelligent systems, ranging from rats to humans and machines, can acquire knowledge about cause-effect relations in novel situations. Decades ago, a number of researchers ([SD88, Sha91]) suggested that causal knowledge can be acquired by a basic learning mechanism, associative learning, that non-human animals commonly employ in classical conditioning paradigms to learn the relationship between stimuli and responses. A major theoretical account of associative learning is the Rescorla-Wagner model, guided by prediction error in updated associative weights on cue-effect links [RW72].

However, subsequent research has produced extensive evidence that human causal learning depends on more sophisticated processes than associative learning of cue-effect links [HC11]; *e.g.*, humans *explore* and *experiment* with dynamic physical scenarios to refine causal hypotheses [BGT18, SF15]. Researchers have demonstrated that humans uncover causal relationships through the discovery of abstract causal structure [WH92] and causal strength [Che97]. Simultaneously, causal graphical models and Bayesian statistical inference have been developed to provide a general representational framework for how causal structure and strength are discovered [GT05, LYL08, GT09, TGK06, BLS15, BDG17, HC11]. Under such a frame-

work, causal connections encode a structural model of the world. States represent some status in the world, and connections between states imply the presence of a causal relationship. However, a critical component in causal learning is active *interaction* with the physical world, based on whether perceived information matches predictions from causal hypotheses. In this chapter, we combine causal learning (a form of model-building) with a model-based planner to effectively achieve tasks in environments where dynamics are unknown.

In contrast to this work beyond the associative account of causal understanding, recent success in the field of deep reinforcement learning (RL) has produced a wide body of research, showcasing agents learning how to play games [MKS15, SHM16, SLA15, SWD17] and develop complex robotic motor skills [LFD16, LHP15]. RL focuses on learning what to do by mapping situations to actions to maximize a reward signal [SB98]. RL has historically been closely linked with associative learning theory and conceives of learning as essentially a process of trial and error. The connection between classical conditioning and temporal-difference learning, a central element of RL, is widely acknowledged [SB90]. Hence, RL could be considered as a modern version of associative learning, where learning is not only guided by prediction error, but also by other learning mechanisms, notably the estimation of the reward function.

Despite this success, the majority of model-free RL methods still have great difficulty transferring learned policies to new environments with consistent underlying mechanics but some dissimilar surface features [ZVM18, KSM17]. This deficiency is due to the limited scope of the agent's overall objective: learning which actions will likely lead to future rewards based on the current state of the environment. In traditional RL architectures, changes to the location and orientation of critical elements (instance-level) in the agent's environment *appear* as entirely new states, even though their functionality often remains the same (in the abstract-level). Since model-free RL agents do not attempt to encode transferable rules governing their environment, new situations appear as entirely new worlds. Although an agent can devise expert-level strategies through experiences in an environment, once that

environment is perturbed, the agent must repeat an extensive learning process to relearn an effective policy in the altered environment.

With these significant developments in RL, is it possible for modern learning models to acquire human-like causal knowledge? To address this question, we designed a novel task to examine learning of action sequences governed by different causal structures, allowing us to determine in what situations humans can transfer their learned causal knowledge. Our design involves two types of basic causal structures (common cause (CC) and common effect (CE); see Fig. 3.2). When multiple causal chains are consolidated into a single structure, they can form either CC or CE schemas. Previous studies using an observational paradigm have found an asymmetry in human learning for common-cause and common-effect structures [WH92].

To design a novel environment for humans, we developed a virtual "escape room". Imagine that you find yourself trapped in an empty room where the only means of escape is through a door that will not open. Although there is no visible keyhole on the door–nor do you see any keys lying around–there are some conspicuous levers sticking out of the walls. Your first instinct might be to pull the levers at random to see what happens, and given the outcome, you might revise your theory about how lever interactions relate to the opening of the door. We refer to this underlying theory as a causal schema: *i.e.*, a conceptual organization of events identified as cause and effect [Hei58]. These schemas are discovered with experience and can potentially be transferred to novel target problems to infer their characteristics [KLH17].

In the escape room example, one method of unlocking the door is to induce the causal schema connecting lever interactions to the door's locking mechanism. However, it remains unclear whether people are equally proficient in uncovering CC and CE schemas in novel situations. In the current study, we first assessed whether human causal learning can be impacted by the underlying structure, comparing learning of a CC structure with learning of a CE structure. We then examined whether learning one type of causal structure can facilitate subsequent learning of a more complex version of the same schema involving a

greater number of causal variables. We compared human performance in a range of learning situations with that of a deep RL model to determine whether behavioral trends can be captured by an algorithm that learns solely by reward optimization with no prior knowledge about causal structure.

After examining whether or not RL can acquire human-level causal knowledge, we present a hierarchical Bayesian learner capable of producing similar learning trends as human learners. For the hierarchical Bayesian learner, the transfer learning problem is viewed as a combination of instance-level associative learning and abstract-level causal learning. We propose: (i) a bottom-up associative learning scheme that determines which attributes are associated with changes in the environment, and (ii) a top-down causal structure learning scheme that infers which atomic causal structures are useful for a task. The outcomes of actions are used to update beliefs about the causal hypothesis space, and our agent learns a dynamics model capable of solving the escape room task.

This chapter integrates multiple modeling approaches to produce a highly capable agent that can learn causal schemas and transfer knowledge to new scenarios. The contributions of this chapter:

1. Showcasing a new environment specifically designed to test causal generalization;
2. Examining how human learners perform in causal generalization tasks;
3. Examining how well model-free RL performs in causal generalization tasks;
4. Learning a bottom-up associative theory that encodes which objects and actions contribute to causal relations;
5. Learning which top-down atomic causal schemas are solutions, thereby learning generalized abstract task structure; and
6. Integrating the top-down and bottom-up learning scheme with a model-based planner to optimally select interventions from causal hypotheses.

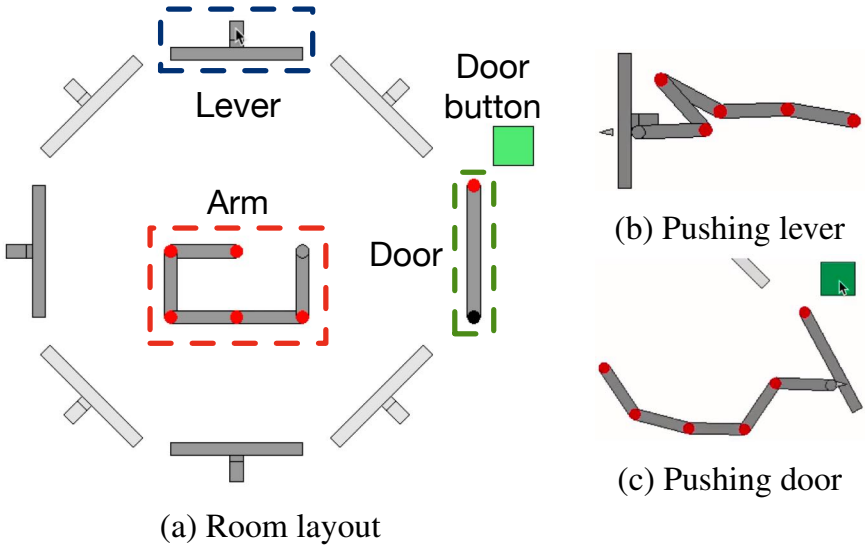


Figure 3.1: (a) Starting configuration of a 3-lever OpenLock room. The arm can interact with levers by either *pushing* outward or *pulling* inward, achieved by clicking either the outer or inner regions of the levers' radial tracks, respectively. Light gray levers are always locked; however, this is unknown to agents. The door can be pushed only after being unlocked. The green button serves as the mechanism to push on the door. The black circle on the door indicates whether or not the door is unlocked; locked if present, unlocked if absent. (b) Pushing on a lever. (c) Opening the door. Copyright reserved to original publication [EMQ20].

The work presented in this chapter was the result of three publications and in collaboration with Colin Summers, Xiaojian Ma, James Kubricht, Siyuan Si, Brandon Rothrock, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu [EKS18, EQZ19, EMQ20]. The authors' contributions include developing the simulation environment, designing and running human experiments, designing and implementing causal theory induction, designing RL experiments, and data analysis. All other portions of the project were not completed by the author.

3.1 OpenLock Task

The OpenLock task, originally presented in [EKS18], requires agents to "escape" from a virtual room by unlocking and opening a door. The door is unlocked by manipulating the levers in a particular sequence (see Fig. 3.1a). Each lever can be manipulated using the robotic arm to *push* or *pull* on levers. Only a subset of the levers, specifically grey levers, are involved in unlocking the door (*i.e.*, active levers). White levers are never involved in unlocking the door (*i.e.*, inactive levers); however, this information is not provided to agents. Thus, at the instance-level, agents are expected to learn that grey levers are always part of solutions and white levers are not. Agents are also tasked with finding *all* solutions in the room, instead of a single solution.

Schemas: The door locking mechanism is governed by two causal schemas: common cause (CC) and common effect (CE). We use the terms common cause 3 (CC3) and common effect 3 (CE3) for schemas with three levers involved in solutions, and common cause 4 (CC4) and common effect 4 (CE4) with four levers; see Fig. 3.2. Three-lever trials have two solutions; four-lever trials have three solutions. Agents are required to find all solutions within a specific room to ensure that they form either CC or CE schema structure; a single solution corresponds to a causal chain.

Constraints: Agents also operate under an action-limit constraint, where only 3 actions (referred to as an *attempt*) can be used to (i) *push* or *pull* on (active or inactive) levers, or (ii) *push* open the door. This action-limit constraint prevents the search depth of interactions with the environment. After 3 actions, regardless of the outcome, the attempt terminates, and the environment resets. Regardless of whether the agent finds all solutions, agents are also constrained to a limited number of attempts in a particular room (referred to as a *trial*; *i.e.*, a sequence of attempts in a room, resulting in finding all the solutions or running out of attempts). An optimal agent will use at most $N + 1$ attempts to complete a trial, where N is the number of solutions in the trial. One attempt would be used to identify the role of

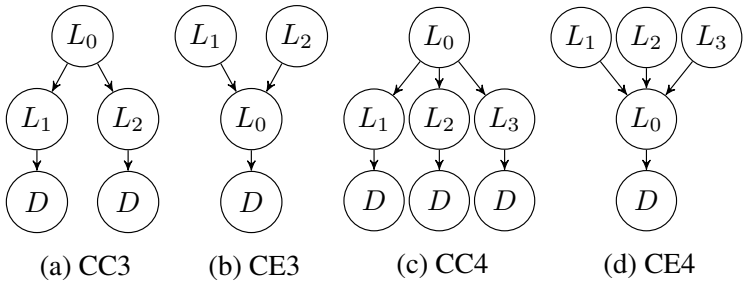


Figure 3.2: (a) common cause 3 (CC3) causal structure. (b) common effect 3 (CE3) causal structure. (c) common cause 4 (CC4) causal structure. (d) common effect 4 (CE4) causal structure. L_0, L_1, L_2 denote different locks, and D the door. Copyright reserved to original publication [EMQ20].

every lever in the abstract schema, and N attempts would be used for each solution.

Training: Training sessions contain only 3-lever trials. After finishing a trial, the agent is placed in another trial (*i.e.*, room) with the *same* underlying causal schema but with a different arrangement of levers. If agents are forming a useful abstraction of task structure, the knowledge they acquired in previous trials should accelerate their ability to find all solutions in the present and future trials.

Transfer: In the transfer phase, we examine agents' ability to generalize the learned abstract causal schema to *different* but similar environments. We use four transfer conditions consisting of (i) congruent cases where the transfer schema adopts the same structure but with an additional lever (CE3-CE4 and CC3-CC4), and (ii) incongruent cases where the underlying schema is changed with an additional lever (CC3-CE4 and CE3-CC4). We compare these transfer results against two baseline conditions (CC4 and CE4), where the agent is trained in a sequence of 4-lever trials.

While seemingly simple, this task is unique and challenging for several reasons. First, requiring the agent to find all solutions rather than a single solution enforces the task as a CC or CE structure, instead of a single causal chain. Second, transferring the agent between trials with the same underlying causal schema but different lever positions encourages

efficient agents to learn an *abstract* representation of the causal schema, rather than learning *instance-level* policies tailored to a specific trial. We would expect agents unable to form this abstraction to perform poorly in any transfer condition. Third, the congruent and incongruent transfer conditions test how well agents are able to adapt their learned knowledge to different but similar causal circumstances. These characteristics of the OpenLock task present challenges for current machine learning algorithms, especially model-free RL algorithms.

3.2 Causal Theory Induction

Causal theory induction provides a Bayesian account of how hierarchical causal theories can be induced from data [GT05, GT09, TGK06]. The key insight is *hierarchy enables abstraction*. At the highest level, a theory provides general background knowledge about a task or environment. Theories consist of principles, principles lead to structure, and structure leads to data. The hierarchy used here is shown in Fig. 3.3a. Our agent utilizes two theories to learn a model of the OpenLock environment: (i) an instance-level associative theory regarding which attributes and actions induce state changes in the environment, denoted as the bottom-up β theory, and (ii) an abstract-level causal structure theory about which atomic causal structures are useful for the task, denoted as the top-down γ theory.

Notation, Definition, and Space: A hypothesis space, Ω_C, is defined over possible causal chains, $c \in \Omega_C$. Each chain is defined as a tuple of subchains: $c = (c_0, \ldots, c_k)$, where k is the length of the chain, and each subchain is defined as a tuple $c_i = (a_i, s_i, cr_i^a, cr_i^s)$. Each a_i is an action node that the agent can execute, s_i is a state node, cr_i^a is a causal relation that defines how a state s_i transitions under an action a_i, and cr_i^s is a causal relation that defines how state s_i is affected by changes to the previous state, s_{i-1}. Each s_i is defined by a set of time-invariant *attributes*, ϕ_i and time-varying *fluents*, f_i [Thi98, Mac42, NC36]; *i.e.*, $s_i = (\phi_i, f_i)$. Action nodes can be directly intervened on, but state nodes cannot. This

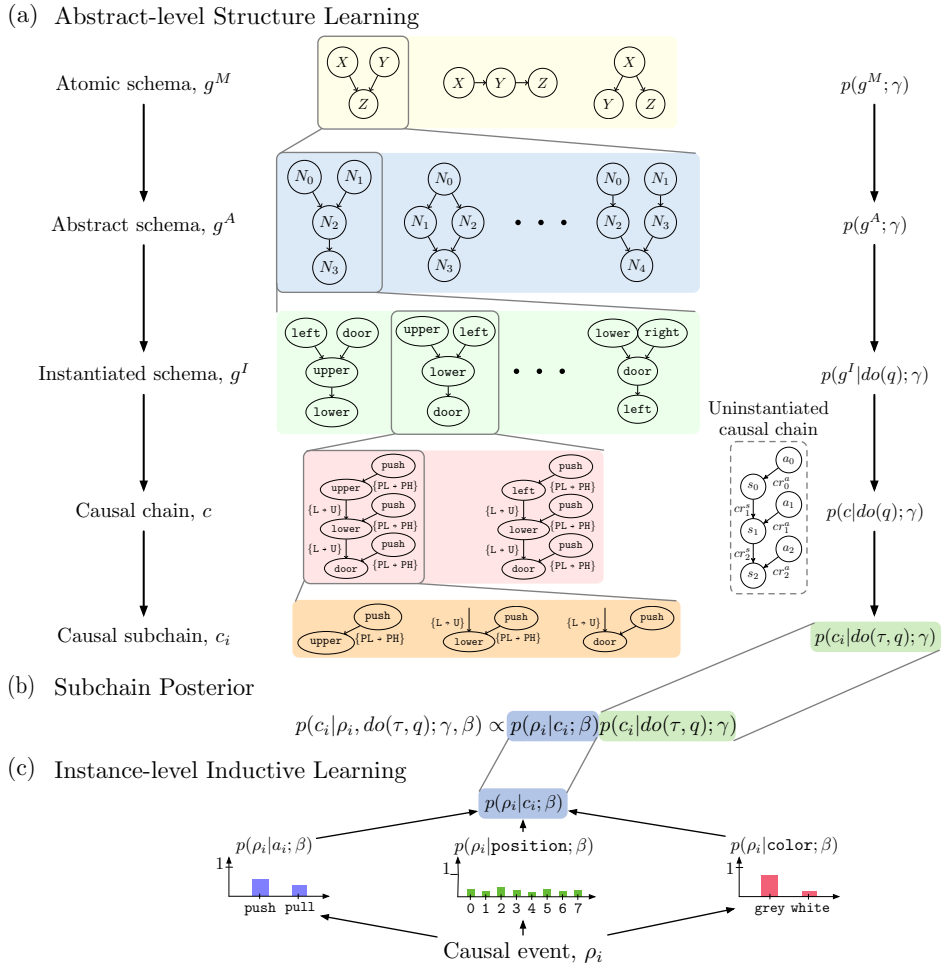


Figure 3.3: Illustration of top-down and bottom-up processes. (a) Abstract-level structure learning hierarchy. Atomic schemas g^M provide the top-level structural knowledge. Abstract schemas g^A are structures specific to a task, but not a particular environment. Instantiated schemas g^I are structures specific to a task and a particular environment. Causal chains c are structures representing a single attempt; an abstract, uninstantiated causal chain is also shown for notation. Each subchain c_i is a structure corresponding to a single action. (b) The subchain posterior is computed using abstract-level structure learning and instance-level inductive learning. (c) Instance-level inductive learning. Each likelihood term is learned from causal events, ρ_i. Copyright reserved to original publication [EMQ20].

means an agent can directly influence (*i.e.*, execute) an action, but how the action affects the world must be *actively* learned. The structure of the general causal chain is shown in the uninstantiated causal chain in Fig. 3.3a. As an example using Fig. 3.1a and the first causal chain in the causal chain level of Fig. 3.3a, if the agent executes *push* on the *upper* lever, the *lower* lever may transition from *pulled* to *pushed*, and the *left* lever may transition from *locked* to *unlocked*.

The space of states is defined as $\Omega_S = \Omega_\phi \times \Omega_F$, where the space of attributes Ω_ϕ consists of position and color, and the space of fluents Ω_F consists of binary values for lever status (*pushed* or *pulled*) and lever lock status (*locked* or *unlocked*). The space of causal relations is defined as $\Omega_{CR} = \Omega_F \times \Omega_F$, capturing the possibly binary transitions between previous fluent values and the next fluent values.

State nodes encapsulate both the time-invariant (attributes) and time-varying (fluents) components of an object. Attributes are defined by low-level features (*e.g.*, position, color, and orientation). These low-level attributes provide general background knowledge about how specific objects change under certain actions; *e.g.*, which levers can be pushed/pulled.

Method Overview: Our agent induces instance-level knowledge regarding which objects (*i.e.*, instances) can produce causal state changes through interaction (see Section 3.2.1) and simultaneously learns an abstract structural understanding of the task (*i.e.*, schemas; see Section 3.2.2). The two learning mechanisms are combined to form a causal theory of the environment, and the agent uses this theory to reason about the optimal action to select based on past experiences (*i.e.*, interventions; see Section 3.2.3). After taking an action, the agent observes the effects and updates its model of both instance-level and abstract-level knowledge.

3.2.1 Instance-level Inductive Learning

The agent seeks to learn which instance-level components of the scene are associated with causal events; *i.e.*, we wish to learn a likelihood term to encode the probability that a causal event will occur. We adhere to a basic yet general associative learning theory: *causal relations induce state changes in the environment, and non-causal relations do not*, referred to as the bottom-up β theory. We learn two independent components: attributes and actions, and we assume they are independent to learn a general associative theory, rather than specific knowledge regarding an exact causal circumstance.

We define Ω_ϕ, the space of attributes, such as position and color, and learn which attributes are associated with levers that induce state changes in the environment. Specifically, an object is defined by its observable features; *i.e.*, the attributes ϕ. We also define Ω_A, a set of actions, and learn a background likelihood over which actions are more likely to induce a state change. We assume attributes and actions are independent and learn each independently.

Our agent learns a likelihood term for each attribute ϕ_{ij} and action a_i using Dirichlet distributions because they serve as a conjugate prior to the multinomial distribution. First, a global Dirichlet parameterized by α^G is used across all trials to encode long-term beliefs about various environments. Upon entering a new trial, a local Dirichlet parameterized by $\alpha^L \in [1, 10]$ is initialized to $k\alpha^G$, where k is a normalizing factor. Such design of using a scaled local distribution is necessary to allow α^L to adapt faster than α^G within one trial; *i.e.*, agents must adapt more rapidly to the current trial compared to across all trials. Thus, we have a set of Dirichlet distributions to maintain beliefs: a Dirichlet for each attribute (*e.g.*, position, and color) as well as a Dirichlet for actions. Similarly, we maintain a Dirichlet distribution over each action a_i to encode beliefs regarding which actions are more likely to cause a state change, independent from any particular circumstance.

We introduce ρ to represent a causal event or observation occurring in the environment.

Our agent wishes to assess the likelihood of a particular causal chain producing a causal event. The agent computes this likelihood by decomposing the chain into subchains

$$p(\rho|c;\beta) = \prod_{c_i \in c} p(\rho_i|c_i;\beta), \tag{3.1}$$

where $p(\rho_i|c_i;\beta)$ is formulated as

$$\begin{aligned}
p(\rho_i|c_i;\beta) &= p(\rho_i|\phi_{i0},\ldots,\phi_{ik},a_i;\beta) && (3.2)\\
&= \frac{p(\phi_{i0},\ldots,\phi_{ik},a_i|\rho_i;\beta)p(\rho_i;\beta)}{p(\phi_{i0},\ldots,\phi_{ik},a_i;\beta)} && (3.3)\\
&= \frac{p(\rho_i;\beta)p(a_i|\rho_i;\beta)\prod_{\substack{\phi_{ij}\in s_i\\ s_i\in c_i}} p(\phi_{ij}|\rho_i;\beta)}{p(a_i;\beta)\prod_{\substack{\phi_{ij}\in s_i\\ s_i\in c_i}} p(\phi_{ij};\beta)} && (3.4)\\
&= \frac{p(\rho_i;\beta)\frac{p(\rho_i|a_i;\beta)p(a_i;\beta)}{p(\rho_i;\beta)}\prod_{\substack{\phi_{ij}\in s_i\\ s_i\in c_i}} \frac{p(\rho_i|\phi_{ij};\beta)p(\phi_{ij};\beta)}{p(\rho_i;\beta)}}{p(a_i;\beta)\prod_{\substack{\phi_{ij}\in s_i\\ s_i\in c_i}} p(\phi_{ij};\beta)} && (3.5)\\
&= \frac{p(\rho_i|a_i;\beta)\prod_{\substack{\phi_{ij}\in s_i\\ s_i\in c_i}} p(\rho_i|\phi_{ij};\beta)}{p(\rho_i;\beta)^k} && (3.6)\\
&\propto p(\rho_i|a_i;\beta)\prod_{\substack{\phi_{ij}\in s_i\\ s_i\in c_i}} p(\rho_i|\phi_{ij};\beta), && (3.7)
\end{aligned}$$

yielding the final derivation

$$p(\rho_i|c_i;\beta) \propto p(\rho_i|a_i;\beta)\prod_{\substack{\phi_{ij}\in s_i\\ s_i\in c_i}} p(\rho_i|\phi_{ij};\beta), \tag{3.8}$$

where k is the number of attributes of the state node s_i in c_i, and $p(\rho_i|\phi_{ij};\beta)$ and $p(\rho_i|a_i;\beta)$ follow multinomial distributions parameterized by a sample from the attribute and action Dirichlet distribution, respectively. We assume $p(\rho_i;\beta)$ is uniform. Note that this derivative is effectively a Naive Bayes approximation of the true joint distribution, $p(\rho_i|\phi_{i0},\ldots,\phi_{ik},a_i;\beta)$.

This scheme combines a set of attributes with a single action but can be easily extended to include multiple actions or additional dimensions to consider for instance-level learning. This knowledge encodes a naive Bayesian view of causal events by independently examining how frequently attributes and actions were involved in causal events.

Intuitively, this bottom-up associative likelihood encodes a naive Bayesian prediction of how likely a particular subchain is to be involved with any causal event by considering how frequently the attributes and actions have been in causal events in the past, without regard for task structure. For example, we would expect an agent in OpenLock to learn that grey levers move under certain circumstances and white levers never move. This instance-level learning provides the agent with task-invariant, basic knowledge about which subchains are more likely to produce a causal effect.

3.2.2 Abstract-level Structure Learning

In this section, we outline how the agent learns abstract schemas; these schemas are used to encode generalized knowledge about task structure that is invariant to a specific observational environment.

A space of atomic causal schemas, Ω_{g^M}, of causal chain, CC, and CE, serve as categories for the Bayesian prior. The belief in each atomic schema is modeled as a multinomial distribution, whose parameters are defined by a Dirichlet distribution. This root Dirichlet distribution's parameters are updated after every trial according to the top-down causal theory γ, computed as the minimal graph edit distance between an atomic schema and the trial's solution structure. This process yields a prior over atomic schemas, denoted as $p(g^M; \gamma)$, and provides the prior for the top-down inference process. Such abstraction allows agents to transfer beliefs between the abstract notions of CC and CE without considering task-specific requirements; *e.g.*, 3- or 4-lever configurations.

Next, we compute the belief in abstract instantiations of the atomic schemas. These abstract schemas share structural properties with atomic schemas but have a structure that matches the task definition. For instance, each schema must have three subchains to account for the 3-action limit imposed by the environment and should have N trajectories, where N is the number of solutions in the trial. Each abstract schema is denoted as g^A, and the space of abstract schemas, denoted Ω_{g^A}, is enumerated. The belief in an abstract causal schema

is computed as

$$p(g^A;\gamma) = \sum_{g^M \in \Omega_{g^M}} p(g^A|g^M)p(g^M;\gamma), \tag{3.9}$$

where $p(g^M;\gamma)$ is the prior over atomic schemas, whose parameters are provided by the atomic schema Dirichlet distribution. The term $p(g^A|g^M)$ is computed as a exponential distribution:

$$p(g^A|g^M) = \frac{1}{Z}\exp(-D(g^A, g^M)), \tag{3.10}$$

where $D(g^A, g^M)$ is the graph edit distance between the abstract schema g^A and the atomic schema g^M, and Z is the normalizing constant, $Z = \sum_{g^A \in \Omega_{g^A}} \exp(-D(g^A, g^M))$.

The abstract structural space can be used to transfer beliefs between rooms; however, we need to perform inference over settings of positions and colors *in this trial* as the agent executes. Thus, the agent enumerates a space of instantiated schemas Ω_{g^I}, where each g^I is an instantiated schema. The agent then computes the belief in an instantiated schema as

$$p(g^I|do(q);\gamma) = \sum_{g^A \in \Omega_{g^A}} p(g^I|g^A, do(q))p(g^A;\gamma), \tag{3.11}$$

where $p(g^I|g^A, do(q))$ is computed as a uniform distribution among all g^I that have $D(g^I, g^A) = 0$ (ignoring vertex labels) *and* contain the solutions found thus far q, and 0 elsewhere. Here, $do(q)$ represents the *do* operator [Pea09], and q represents the solutions already executed. Conditioning on $do(q)$ constrains the space to have instantiated solutions that contain the solutions already discovered by the agent in this trial.

Causal chains c define the next lower level in the hierarchy, where each chain corresponds to a single attempt. The belief in a causal chain is computed as

$$p(c|do(q);\gamma) = \sum_{g^I \in \Omega_{g^I}} p(c|g^I, do(q))p(g^I|do(q);\gamma), \tag{3.12}$$

where $p(c|g^I, do(q))$ is uniform across all $c \in g^I$ and 0 elsewhere.

Finally, the agent computes the belief in each possible subchain as

$$p(c_i|do(\tau, q);\gamma) = \sum_{c \in \Omega_C} p(c_i|c, do(\tau, q))p(c|do(q);\gamma), \tag{3.13}$$

where $do(\tau, q)$ represents the intervention of performing the action sequence executed thus far in this attempt τ, and performing all solutions found thus far q. The term $p(c_i|c, do(\tau, q))$ is uniform across all $c_i \in c$ and 0 elsewhere. This hierarchical process allows the agent to learn and reason about abstract task structure, taking into consideration the specific instantiation of the trial, as well as the agent's history within this trial.[0]

Additionally, if the agent encounters an action sequence that does not produce a causal event, the agent prunes all chains that contain the action sequence from Ω_C and prunes all instantiated schemas that contain the corresponding chain from Ω_{g^I}. This pruning strategy means the agent assumes the environment is deterministic and updates its theory about which causal chains are causally plausible through interactions on the fly.

3.2.3 Intervention Selection

Our agent's goal is to pick the action it believes has the highest chance of (i) being causally plausible in the environment *and* (ii) being part of the solution to the task. We decompose each subchain c_i into its respective parts, $c_i = (a_i, s_i, cr_i^a, cr_i^s)$. The agent combines the top-down and bottom-up processes into a final subchain posterior:

$$p(c_i|\rho_i, do(\tau, q); \gamma, \beta) \propto p(\rho_i|c_i; \beta)p(c_i|do(\tau, q); \gamma). \tag{3.14}$$

Next, the agent marginalizes over causal relations and states to obtain a final, action-level term to select interventions:

$$p(a_i|\rho_i, do(\tau, q); \gamma, \beta) = \sum_{s_i \in \Omega_S} \sum_{cr_i^a \in \Omega_{CR}} \sum_{cr_i^s \in \Omega_{CR}} p(a_i, s_i, cr_i^a, cr_i^s|\rho_i, do(\tau, q); \gamma, \beta). \tag{3.15}$$

The agent uses a model-based planner to produce action sequences capable of opening the door (following human participant instructions in [EKS18]). The goal is defined as reaching a particular state s^*, and the agent seeks to execute the action a_t to maximize the posterior subject to the constraints that the action appears in the set of chains that satisfy the goal, $\Omega_{C^*} = \{c \in \Omega_C \mid s^* \in c\}$. We define the set of actions that appear in chains satisfying the

goal as $\Omega_{A^*} = \{a \in \Omega_A | \exists c \in \Omega_{C^*}, \exists\ s, cr^a, cr^s\ | (a, s, cr^a, cr^s) \in c\}$. The agent's final planning goal is

$$a_t^* = \arg\max_{a_i \in \Omega_{A^*}} p(a_i | \rho_i, do(\tau, q); \gamma, \beta). \tag{3.16}$$

At each time step, the agent selects the action that maximizes this planning objective and updates its beliefs about the world as described in Section 3.2.1 and Section 3.2.2. This iterative process consists of optimal decision-making based on the agent's current understanding of the world, followed by updating the agent's beliefs based on the observed outcome.

3.3 Materials and Methods

Next, we illustrate experimental setups for three classes of experiments: (1) human subjects, (2), our causal theory induction learner, and (3) reinforcement learning (RL).

3.3.1 Human Subject Experimental Setup

A total of 240 undergraduate students (170 female; mean age = 21.2) were recruited from the University of California, Los Angeles (UCLA) Department of Psychology subject pool and were compensated with course credit for their participation. Participants were not explicitly told which levers were active or inactive but were instead required to learn the distinction through trial and error. This was not generally difficult, however, as the inactive levers could never be moved. The order in which the active levers needed to be moved followed either a common cause (CC) or common effect (CE) schema (see Fig. 3.2), and participants were given 30 attempts to discover *every* solution in each situation. Participants were instructed to consider solutions as "combinations" to each lock, and discovery of every solution/combination was required to ensure that participants understood the underlying causal schema in each situation. Participants also operated under a movement-limit constraint whereby only three movements could be used to both (1) interact with the levers (two movements) and (2) push open the door (one movement). If a participant tried to move an active lever in an incorrect

order, the lever would remain stationary and a movement would be expended. Each trial reverted to its initial state once the three movements were expended, and the experiment automatically proceeded to the next trial after 30 attempts. The number of remaining solutions and attempts were provided in a console window located on the same screen as the OpenLock application.

In the environment, users commanded the movement of a simulated robot arm by clicking on desired elements in a 2D display. Levers could either be pushed or pulled by clicking on their inner or outer tracks, although pulling on a lever was never required to unlock the door. There were either 3 or 4 active levers in each lock situation. We refer to the 3- and 4-lever common cause situations as CC3 and CC4 (Fig. 3.2(a), Fig. 3.2(b)), respectively, and the 3- and 4-lever common effect situations as CE3 and CE4 (Fig. 3.2(c), Fig. 3.2(d)), respectively. Note that these numbers correspond with the number of *active* levers. The status of the door (*i.e.*, either locked or unlocked) was indicated by the presence or absence of a black circle located opposite the door's hinge. Once the door was unlocked and the black circle disappeared, participants could command the robot arm to push the door open by clicking on a green *push* button. The robot arm consisted of five segments that were free to rotate such that all elements in the display were easily reached by the arm's free end; the arm position control was implemented using inverse kinematics. Box2D [Cat11] was used to handle collision, and the underlying simulation environment uses OpenAI Gym [BCP16] as the virtual playground to train agents and enforce causal schemas through a finite state machine.

Participants were randomly assigned to one of six conditions in a between-subjects experimental design (40 participants per condition) and began the experiment by viewing a set of instructions outlining important components and details in the lock environment[1]. Fifteen additional participants were recruited but subsequently removed from the analysis due to their inability to complete any trial in the allotted number of attempts. The first

[1]The instructional video can be viewed at `https://vimeo.com/265302423`

two experimental conditions were baselines that contained five different lock situations comprised of either CC4 or CE4 trials, exclusively. These baseline conditions for the two control groups, denoted as CC4 and CE4, were included to assess whether human causal learning can be impacted by the underlying structure, comparing learning of a common-cause structure with learning of a common-effect structure. For the remaining four conditions, we examined whether learning one type of causal structure can facilitate subsequent learning of a more complex version of the same schema involving a greater number of causal variables (*i.e.*, active levers). The four conditions contained six training trials with 3-lever situations, followed by one transfer trial with a 4-lever situation. The schema underlying the 3- and 4-lever situations was either congruent (CC3-CC4, CE3-CE4) or incongruent (CC3-CE4, CE3-CC4) and remained the same throughout the 3-lever training trials. Participants required approximately 17.4 min to complete the baseline trials and 17.3 min to complete the training and transfer trials.

3.3.2 Causal Theory Induction Experimental Setup

The causal theory induction-based learner was run in a manner identical to the human subjects. We executed 40 agents in each condition, matching the number of human subjects described in Section 3.3.1. This allows us to directly compare the results of the causal theory induction-based learner with human subject results.

3.3.3 Reinforcement Learning Experimental Setup

Given the success of model-free RL, we seek to test the limits of these models by examining if they are capable of solving our causal generalization task. In RL experiments, we want to answer:

1. Can predominate, state-of-the-art model-free RL algorithms solve the OpenLock task?
2. What transferable representations, if any, do these RL agents establish?

Notice our task definition requires agents to find *all* solutions in a trial. This requirement means that an agent that *memorizes* and biases to one specific solution will perform poorly in unseen rooms. Agents must form abstract transferable notions of the task or must memorize all possible settings of the task, including unseen settings, to perform well.

To answer the first question, we show the performance of model-free RL algorithms. We try to improve their performances by providing several reward strategies. The details of algorithms, tasks, and rewards we used can be found in Section 3.3.3.1.

To answer the second question, if the agents are able to establish such concepts, they can master the task with similar causal schema both better and faster than training on that task from scratch; *i.e.*, we expect to see a positive transfer. In this experiment, all the agents are first trained on 3-lever tasks, then we transfer these agents to target 4-lever tasks using fine-tuning. By comparing the results in our transfer experiments with directly training on target tasks (*i.e.*, baseline experiments), we can verify whether the agents are able to build such abstract casual concepts.

For RL, there a few specifics of the OpenLock environment that are pertinent. Readers are encouraged to examine [EKS18] for additional details.

- **State Space:** The state space consists of 16 binary dimensions: 7 for the state of each lever (*pushed* or *pulled*), 7 dimensions for the color of each lock (*grey* or *white*), 1 dimension for the state of the lock (*locked* or *unlocked*), and 1 dimension for the state of the door (*closed* or *open*).
- **Action Space:** The action space is a discrete space with 15 dimensions: each of the 7 levers has 2 actions (*push* and *pull*), and the door has one action (*push*).

3.3.3.1 Algorithms, causal schemas and Rewards

We select a set of predominate, state-of-the-art RL algorithms as baselines, including deep Q-network (DQN) [MKS15], DQN with prioritized experience replay (DQN-PE) [SQA16],

advantage actor-critic (A2C) [MBM16], trust region policy optimization (TRPO) [SLA15], proximal policy optimization (PPO) [SWD17] and model-agnostic meta learning (MAML) [FAL17]. Table 3.1 lists all the baselines we considered. These algorithms have been applied to solve a variety of tasks including Atari Games [MKS15], classic control, and even complex visual-motor skills [LPK18], and they have shown remarkable performance on these tasks when large amounts of simulated or real-world exploration data are available.

When executing various model-free RL agents under the same experimental setup as human learners, no meaningful learning takes place. Instead, we train RL agents by looping through all rooms repeatedly (thereby seeing each room multiple times). Agents are also allowed 700 attempts in each trial to find all solutions. During training, agents execute for 200 training iterations, where each iteration consists of looping through all six 3-lever trials. During transfer, agents execute for 200 transfer iterations, where each iteration consists of looping through all five 4-lever trials. Note that the setup for RL agents is advantageous; in comparison, both the proposed model and human subjects are only allowed 30 attempts (versus 700) during the training and 1 iteration (versus 200) for transfer.

RL agents operate directly on the state of the simulator encoded as a 16-dimensional binary vector consisting of:

1. Status of each of the 7 levers (*pushed* or *pulled*),
2. Color of each of the 7 levers (*grey* or *white*),
3. Status of the door (*open* or *closed*), and
4. Status of the door lock indicator (*locked* or *unlocked*).

The 7-dimensional encoding of the status and color of each lever encodes the position of each lever; *e.g.*, the 0^{th} index corresponds to the upper-right position. Despite direct access to the simulator's state, RL approaches were unable to form a transferable task abstraction.

For baseline experiments (To answer Q1 in Section 3.3.3)	For transfer experiments (To answer Q2 in Section 3.3.3)
DQN on 3-lever task from scratch	Fine-tune DQN on 4-lever task
DQN-PE on 3-lever task from scratch	Fine-tune DQN-PE on 4-lever task
A2C on 3-lever task from scratch	Fine-tune A2C on 4-lever task
TRPO on 3-lever task from scratch	Fine-tune TRPO on 4-lever task
PPO on 3-lever task from scratch	Fine-tune PPO on 4-lever task
MAML (Meta learning with 3 and 4-lever tasks)	MAML (N shot adaption on 4-lever task)

Table 3.1: Baselines used in our experiments.

Additionally, we also include a baseline of MAML [FAL17]. Note that the MAML does not employ a standard transfer learning setting as it requires to access the target task during the meta-learning phase, which can be more advantageous than other transfer methods. Our main goal is to verify whether the state-of-the-art meta-learning algorithm (*i.e.*, MAML) can solve the OpenLock task by forming the correct causal abstraction of the task.

A reward function that only rewards for unique solutions performed best, meaning agents were only rewarded the *first* time they found a particular solution. This is similar to the human experimental setup, under which participants were informed when they found a solution for the first time (thereby making progress towards the goal of finding *all* solutions) but were not informed they executed the same solution multiple times (thereby not making progress towards the goal).

We utilized a plethora of reward functions to explore under what circumstances these RL approaches may succeed. Our agents used sparse reward functions, shaped reward functions, and conditional reward functions that encourage agents to find unique solutions. Rewards in the OpenLock environment are very sparse; agents must search in a large space of possible attempts (*i.e.* action sequences) of which there are 2 or 3 action sequences that achieve the

task. Sparse rewards have traditionally been a challenge for RL [SB98]. To overcome this, we enhance the reward by shaping it to provide better feedback for the agent; we introduce task-relevant penalties and bonuses. We utilize 6 reward strategies:

Basic (B) The agent will receive a reward for unlocking the door and will receive the largest reward for opening the door. No other rewards are granted for all other outcomes.

Unique Solution (U) Inherits from Reward B, but the agent only receives a reward when unlocking/opening the door with a new solution. There are a finite number of solutions (2 for 3-lever trials and 3 for 4-lever trials). This reward is designed to encourage the agent to find all solutions within a trial, instead of only finding/pursuing the first solution found.

Reward B and Negative Immovable (B+N) Inherits from Reward B, but introduces an extra penalty for manipulating an immovable lever (Reward N). This is judged by whether a state change occurs after executing an action; this penalty is designed to encourage the agent to only interact with movable levers.

Reward U and Negative Immovable (U+N) This reward is a combination of Reward U and the Negative Immovable penalty (Reward N) introduced in Reward B+N.

Reward N and Solution Multiplier (N+M) This reward inherits from Reward B, but in this reward setting, we encourage the agent to find more solutions in a slightly different way from Reward U. Instead of only providing reward when finishing the task with a new solution, the agent will receive a reward every time it unlocks/opens the door, but when the agent finds a unique solution, the reward it receives is multiplied by a fixed factor (> 1). This effectively encourages the agent to find new solutions in a more reward-dense setting. In addition, we also use the Negative Immovable penalty (Reward N) for learning efficiency.

Reward N+M and Partial Sequence (N+M+P) Inherits from Rewards B, N, and M, but adds a Partial Sequence bonus. When the executed action sequence is exactly a prefix of a solution to the current trial (no matter whether this solution has been found out or not), the agent will receive a bonus. This is a form of reward shaping to overcome the sparse reward problem.

3.3.4 Hyper-parameters and Training Details

Table 3.2 presents the hyperparameters and training details for our experiments. We selected these parameters through several preliminary experiments.

3.3.4.1 Experimental Procedure

Here we describe the complete experimental procedures for RL agents. Each agent is trained for 200 *iterations*. In each iteration, there are six *trials* for 3-lever tasks (CC3 and CE3; referred to as the training phase) and five for 4-lever tasks (CC4 and CC4; referred to as the testing phase). Agents are allowed to take at most 700 *attempts* to find all of the solutions within a trial. A typical trial proceeds as follows:

1. A new trial starts.
2. Agent is permitted a finite number of attempts to find all solutions. An attempt will start from the initial state of the environment, and end with opening the door or reaching the maximum action limit.
3. A trial ends either when all the solutions have been found or the agent reaches the maximum attempt limit.
4. After finding all solutions or running out of attempts, the agent is placed in the next trial with *different* lever configurations but the same causal schema during the training phase.

5. After completing all trials in the training phase, the agent is placed into a single 4-lever trial for the testing phase.

There are six configurations in total for 3-lever tasks and five configurations for 4-lever tasks. The configuration of the lever is selected in a loop; the initial order of the configurations is randomized per agent, but each agent sees the same room ordering for the entire experiment.

We evaluate the final performances after all iterations are finished. The details of the evaluation are discussed in Section 3.3.4.1.

Evaluation Details We expect an agent learning the correct abstractions and generalizations to quickly adapt to similar but slightly different circumstances. More specifically, an agent learning the correct abstractions should perform better (*i.e.* have fewer attempts) as the agent encounters more trials with the same causal schema. We propose several criteria to evaluate agents (see supplementary of [EMQ20] for further details):

- **Attempt Amount** This curve shows the number of attempts used in each trial. Because a trial terminates when all solutions have been found, an agent with better performance will have fewer attempts per trial. Moreover, the decreasing speed of this curve can also show how quickly the agent mastered finding all solutions.

- **Percentage of Found Solution** This curve shows how many solutions the agent found within a trial, *e.g.*, if the agent found all the 3 solutions (for a 4-lever task), this value will be 1 for this trial. This plot also shows how well the agent mastered find all solutions.

- **Averaged Trial Reward** This curve shows the averaged reward in a trial (reward sum divided by the number of attempts). Since the reward strategies are varied in our experiments, this value cannot be a direct criterion to compare the performance of various experimental settings.

Baseline Experiments In baseline experiments, we want to evaluate the agents' performance on a single causal schema. The agent needs to do several trials successively. Among these trials, the causal schema is fixed, while the lever configurations and observational solutions are varied (structurally, the solutions remain the same). The goal in each trial is to find all the solutions using as few attempts as possible. We evaluate all the 5 algorithms (DQN, DQN-PE, A2C, TRPO and PPO) on four causal schemas, and the tabulated results are presented in the supplementary material of [EMQ20].

In general, 3-lever tasks are easier than 4-lever tasks, because there are more solutions to find in the latter case. Specifically, for rewards that do not encourage finding multiple solutions, such as Reward B and N, it is quite difficult for agents to find all the solutions, and agents are frequently biased to one specific solution. In other words, agents *memorize* a single solution instead of learning the abstract, multi-solution causal schema. As for the reward strategies that encourage finding multiple solutions, Reward U is the best for most of the agents. In addition, for some importance sampling-based policy gradient methods (PPO/TRPO), an extra penalty (Reward N) can slightly improve the stability and final results.

In the Reward N+M and Reward N+M+P strategies, we introduce some reward shaping techniques, including reward multiplier and partial sequence bonus, to mitigate the sparse reward problem. However, the results are worse and more unstable. We posit that this may be caused by the positive reward for non-unique solutions. Although the agents are encouraged to find new solutions using the multiplied reward, nothing prevents agents from being biased towards a specific solution, yielding a sub-optimal policy. To eliminate this, we may need to adjust the learning rate dynamically as solutions are found. Thus, selecting hyper-parameters for the last 2 reward strategies is challenging, and the results are difficult to match expectations.

Another interesting result is the performance of value-based methods (DQN, DQN-PE). For all causal schemas and reward strategies, these methods do not perform well under any

of our experiments. Since the lever settings vary between trials, it is extremely difficult for the agent to build a universal value function based on discrete state-action input [EKS18]. The causal schema remains the same, but the value function learned is not directly based on the abstract causal state. The RL agents examined do not appear able to construct a representation capable of inferring the connection between the explicit discrete state and the abstract causal state.

Transfer Experiments In transfer experiments, we first train our agents in a 3-lever task and then to a 4-lever task. We perform quantitative evaluations on the target 4-lever task for all the transferred models. Additionally, we also compare them with the models that trained on a 4-lever task from scratch (*i.e.*; baseline experiments). If the agents form useful abstract structural representations of tasks, we expect them to complete the 4-lever task faster than training from scratch. All five algorithms and six reward strategies are considered. The results are listed in the supplementary material of [EMQ20].

Reward strategies that were not effective in baseline experiments were also not effective in transfer experiments, as expected. Baseline experiments showed that policy-based methods (A2C, PPO, TRPO) with explicit encouragement to multi-solution performed better; these agents mastered most of the solutions. As mentioned above, if an agent is able to establish a concept to the corresponding causal schema, it should have comparable transfer performance regarding the performance of agent training on a 4-lever task from scratch, and it is also expected to converge faster. However, for both CC4 and CE4 causal schemas, there is a significant gap between transfer performance and training performance. Even under the most effective reward strategies (Reward U, Reward U+N, and Reward N+M), the agents find it hard to match the corresponding training performance, indicating negative transfer.

3.4 Results

3.4.1 Human Subject and Model Results

The results using the proposed model are shown in Fig. 3.4. These results are qualitatively and quantitatively similar to the human participant results presented in [EKS18], and starkly different from the RL results in Section 3.4.2

Our agent does not require looping over trials multiple times; it is capable of learning and generalizing from seeing each trial only one time. In the baseline agents, the CE4 condition was more difficult than CC4; this trend was also observed in human participants. During transfer, we see a similar performance as the baseline results; however, congruent cases (transferring from the same structure with an additional lever) were easier than incongruent cases (transferring to a different structure with an additional lever; CE4 transfer); this result was statistically significant for CE4: $t(79) = 3.0$; $p = 0.004$. For CC4 transfer, no significance was observed ($t(79) = 0.63$; $p = 0.44$), indicating both CC3 and CE3 obtained near-equal performance when transferred to CC4.

These learning results are significantly different from the RL results; the proposed causal theory-based model is capable of learning the correct abstraction using instance and structural learning schemes, showing similar trends as the human participants. It is worth noting that RL agents were trained under highly advantageous settings. RL agents: (i) were given more attempts per trial; and (ii) more importantly, were allowed to learn in the same trial multiple times. In contrast, the present model learns the proper mechanisms to: (i) transfer knowledge to structurally equivalent but observationally different scenarios (baseline experiments); (ii) transfer knowledge to cases with structural differences (transfer experiments); and (iii) do so using the *same experimental setup* as humans. The model achieves this by understanding which scene components are capable of inducing state changes in the environment while leveraging overall task structure.

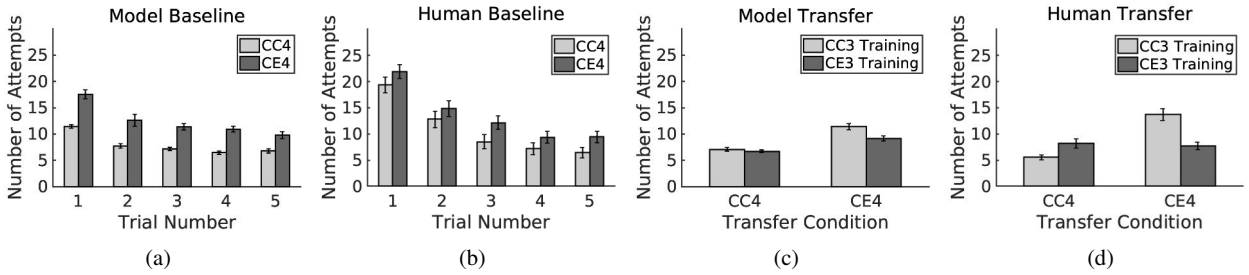
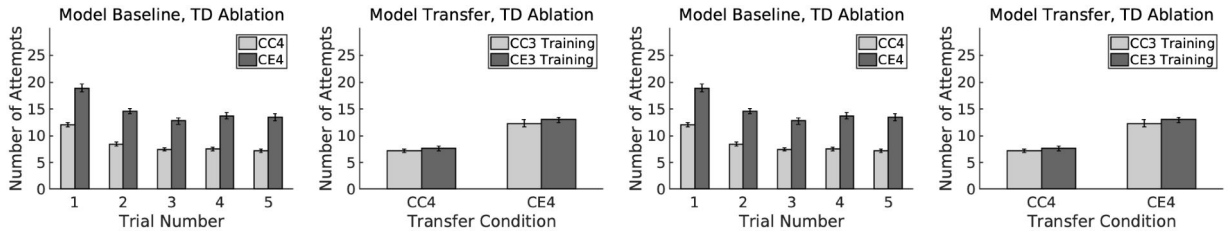


Figure 3.4: Model performance *vs.* human performance. (a) Proposed model baseline results for CC4/CE4. We see an asymmetry between the difficulty of CC and CE. (b) Human baseline performance. (c) Proposed model transfer results for training in CC3/CE3. The transfer results show that transferring to an incongruent CE4 condition (*i.e.*, different structure, additional lever; *i.e.*, CC3 to CE4) was more difficult than transferring to a congruent condition (*i.e.*, same structure, additional lever; *i.e.*, CE3 to CE4). However, the agent did not show a significant difference in difficulty when transferring to congruent or incongruent condition for the CC4 transfer condition. (d) Human transfer performance. Copyright reserved to original publication [EMQ20].

3.4.1.1 Model Ablation Results

In this section, we present additional results from our proposed method. Specifically, we show how well the model performs under two ablations: (i) top-down structure learning and (ii) bottom-up instance learning. This examination seeks to identify to what degree and how well much each model component contributes to the model's performance. In our formulation, these ablations amount to setting a probability of 1 for the ablated component in the subchain posterior; *i.e.*, the subchain posterior reduces to the remaining active model component (bottom-up during a top-down ablation and top-down during a bottom-up ablation).

Figure 3.5 shows the results of the ablated model. In Figure 3.5a and Figure 3.5b, the model is ablated to disable the top-down abstract structure learning. We see the agent performing with similar trends as the full model results, but with worse performance. This is due to the agent learning the bottom-up associative theory regarding which instances can be



(a) (b) (c) (d)

Figure 3.5: Results using the proposed theory-based causal transfer under ablations. (a) Proposed model baseline results under a top-down ablation (*i.e.*, only instance-level learning occurred). (b) Proposed model transfer results under a top-down ablation. (c) Proposed model baseline results under a bottom-up ablation (*i.e.*, only abstract-level structure learning occurred). (b) Proposed model transfer results under a bottom-up ablation. Copyright reserved to original publication [EMQ20].

manipulated to produce a causal effect, but the agent performs worse due to the lack of task structure. During transfer, we see little difference (with no significance; $t(79) = 0.8; p = 0.42$ and $t(79) = 0.8; p = 0.43$ for CC4 and CE4 respectively) between the training groups. This is expected; an agent that learns no task structure should exhibit no difference between tasks. This agent is essentially aimlessly searching the structure space, biased towards *any* structure with subchains with a high likelihood of producing a causal event.

Figure 3.5c and Figure 3.5d show the model ablated with the bottom-up instance learning disabled. In the baseline results, we see a slight increase in performance over time for CC4; this is because the agent is becoming more confident in which structure governs the environment. However, this version of the model has no regard for whether or not an agent can interact with a particular instance (*i.e.*, it lacks the bottom-up associative theory regarding causal events). Because of this limitation, the agent must try many possible instantiations of the correct abstract structure before finding a solution. During transfer, we see the agent benefiting most from training in CC3, which is counter-intuitive for the CE4 transfer condition.

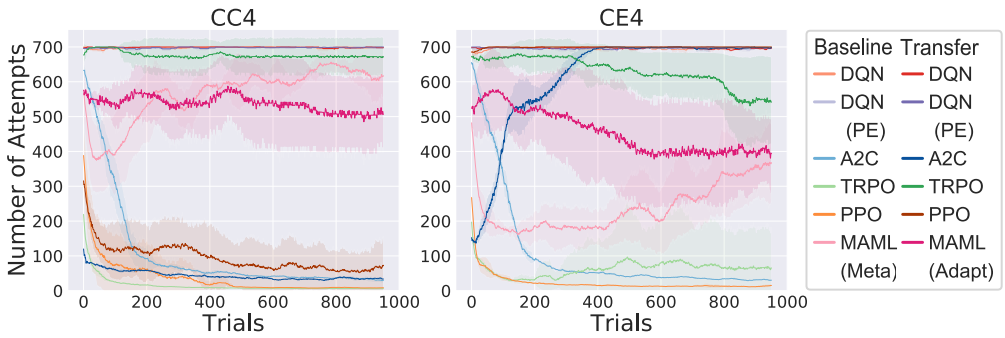


Figure 3.6: RL results for baseline and transfer conditions. Baseline (no transfer) results show the best-performing algorithms (PPO, TRPO) achieving approximately 10 and 25 attempts by the end of the baseline training for CC4 and CE4, respectively. A2C is the only algorithm to show positive transfer; A2C performed better with training for the CC4 condition. The last 50 iterations are not shown due to the use of a smoothing function. Copyright reserved to original publication [EMQ20].

However, we believe this is best explained from a decision tree perspective, as elaborated in the main text. Throughout all model and human experiments, we observed that CE was more difficult than CC. From a decision tree perspective, agents that learn a CC structure will first identify the first lever in the structure; this is the only lever they can interact with initially. After identifying this lever, they can then push on either remaining lever to unlock the door. While this strategy will not work for CE directly, it may still benefit an agent only equipped with structure learning. For instance, when applying this strategy to CE, the agent may find the *first* solution faster. After finding the first solution, the space of second solutions is constrained to contain the first solution. From here, despite having learned the "wrong" structure for this task, the agent may find both remaining solutions faster. This is an unexpected phenomenon and will be examined in future work.

3.4.2 Reinforcement Learning Results

The model-free RL results, shown in Fig. 3.6, demonstrate that A2C, TRPO, and PPO are capable of learning how to solve the OpenLock task from scratch. However, A2C in the CC4 condition is the only agent showing positive transfer; every other agent in every condition shows negative transfer.

These results indicate that current model-free RL algorithms are capable of learning how to achieve this task; however, the capability to transfer the learned abstract knowledge is markedly different compared to human performance in [EKS18]. Due to the overall negative transfer trends shown by nearly every RL agent, we conclude that these RL algorithms cannot capture the correct abstractions to transfer knowledge between the 3-lever training phase and the 4-lever transfer phase. Note that the RL algorithms found the CE4 condition more difficult than CC4, a result also shown in our proposed model results and human participants.

Empirical results of MAML Here we separately present the empirical results of MAML since it is a meta-learning approach that does not come from the same category as other transfer learning methods (see Table 3.1). We conduct experiments on MAML with only the reward strategy of *unique solutions* (**Reward U**) as this strategy overall provides the best performances. All the numerical results are presented in the supplementary material of [EMQ20].

As the meta optimizer we use in MAML is TRPO [SLA15], we compare the adaption results with TRPO in transfer experiments on CC4/CE4, which can be found in the supplementary material of [EMQ20]. The results indicate that during the few-shot adaption phase, MAML overall outperforms than fine-tuning policy previously learned on a 3-lever task with TRPO, which demonstrates that the transferring, or adaption do benefit from meta-learning from both the 3 and 4-lever tasks. However, when comparing with the oracle baseline results that directly training on 4-lever tasks, there is still a significant performance

gap, which indicates that the MAML agent cannot master the target tasks well. Namely, being similar to all the fine-tuning methods, meta-learning on the previous task with the same causal schema can improve neither the performances of subsequent policy learning on target task nor the convergence properties but misleads the policy learning even with similar causal schema. This demonstrates that the state-of-the-art meta-learning approach also may not be able to establish a useful concept toward the causal schemas among the tasks it encounters during the meta-learning phase.

3.5 Conclusion and Discussion

In this chapter, we examined how humans solved a causal generalization task and showed how the theory-based causal transfer coupled with an associative learning scheme can be used to learn transferable structural knowledge under both observationally and structurally varying tasks. We executed a plethora of model-free RL algorithms, none of which learned a transferable representation of the OpenLock task, even under favorable baseline and transfer conditions. In contrast, the proposed model results are not only capable of successfully completing the task but also adhere closely to the human participant results in [EKS18].

These results suggest that current model-free RL methods lack the necessary learning mechanisms to learn generalized representations in hierarchical, structured tasks. Our model results indicate human causal transfer follows similar abstractions as those presented in this work, namely learning abstract causal structures and learning instance-specific knowledge that connects this particular environment to abstract structures. The model presented here can be used in any reinforcement learning environment where: (i) the environment is governed by a causal structure, (ii) causal cues can be uncovered from interacting with objects with observable attributes, and (iii) different circumstances share some common causal properties (structure and/or attributes).

3.5.1 Discussion

Why is causal learning important for RL? We argue that causal knowledge provides a succinct, well-studied, and well-developed framework for representing cause and effect relationships. This knowledge is invariant to extrinsic rewards and can be used to accomplish many tasks. In this work, we show that leveraging abstract causal knowledge can be used to transfer knowledge across environments with similar structure but different observational properties.

How can RL benefit from structured causal knowledge? Model-free RL is apt at learning a representation to maximize a reward within simple, non-hierarchical environments using a greedy process. Thus, current approaches do not restrict or impose learning an abstract structural representation of the environment. RL algorithms should be augmented with mechanisms to learn explicit structural knowledge and jointly optimized to learn both an abstract structural encoding of the task while maximizing rewards.

Why is CE more difficult than CC? Human participants, RL, and the proposed model all found CE more difficult than CC. A natural question is: why? We posit that it occurs from a decision-tree perspective. In the CC condition, if the agent makes a mistake on the first action, the environment will not change, and the rest of the attempt is bound to fail. However, should the agent choose the correct grey lever, the agent can choose either of the remaining grey levers; both of which will unlock the door. Conversely, in the CE condition, the agent has two grey levers to choose from in the first action; both will unlock the lever needed to unlock the door. However, the second action is more ambiguous. The agent could choose the correct lever, but it could also choose the other grey lever. Such complexity leads to more failure paths from a decision-tree planning perspective. The CC condition receives immediate feedback on the first action as to whether or not this plan will fail; the CE condition, on the other hand, has more failure pathways. We plan to investigate this property further, as this asymmetry was unexpected and unexplored in the literature.

What other theories may be useful for learning causal relationships? In this work, we adhere to an associative learning theory. We adopt the theory that *causal relationships induce state changes*. However, other theories may also be appealing. For instance, the associative theory used does not directly account for long-term relationships (delayed effects). More complex theories could potentially account for delayed effects; *e.g.*, when an agent could not find a causal attribute for a particular event, the agent could examine attributes jointly to best explain the causal effect observed. Prior work has examined structural analogies [HF11, ZGJ19, ZJG19] and object mappings [FGT18] to facilitate transfer; these may also be useful to acquire transferable causal knowledge.

How can hypothesis space enumeration be avoided? Hypothesis space enumeration can quickly become intractable as problems increase in size. While this work used a fixed, fully enumerated hypothesis space, future work will include examining how sampling-based approaches can be used to iteratively generate causal hypotheses. [BDG17] showed a Gibbs-sampling based approach; however, this sampling should be guided with top-down reasoning to guide the causal learning process by leveraging already known causal knowledge with proposed hypotheses.

How well would model-based RL perform in this task? Model-based RL may exhibit faster learning within a particular environment but still lacks mechanisms to form abstractions that enable human-like transfer. This is an open research question, and we plan on investigating how abstraction can be integrated with model-based RL methods.

How is this method different from hierarchical RL? Typically, hierarchical RL is defined on a hierarchy of goals, where subgoals represent *options* that can be executed by a high-level planner [CBS05]. Each causally plausible hypothesis can be seen as an option to execute. This work seeks to highlight the importance of leveraging causal knowledge to form a world model and using said model to guide a reinforcement learner. In fact, our work can be recast as a form of hierarchical model-based RL.

Future work should primarily focus on how to integrate the proposed causal learning algo-

rithm directly with reinforcement learning. An agent capable of integrating causal learning with reinforcement learning could generalize world dynamics (causal knowledge) and goals (rewards) to novel but similar environments. One challenge, not addressed in this paper, is how to generalize rewards to varied environments. Traditional reinforcement learning methods, such as Q-learning, do not provide a mechanism to extrapolate internal values to similar but different states. In this work, we showed how extrapolating causal knowledge can aid in uncovering causal relationships in similar environments. Adopting a similar scheme for some form of reinforcement learning would enable reinforcement learners to succeed in the OpenLock task without iterating over the trials multiple times, and could enable one-shot reinforcement learning. Future work will also examine how a learner can iteratively grow a causal hypothesis while incorporating a background theory of causal relationships.

Table 3.2: Hyperparameters and training details. See supplementary material of [EMQ20] for additional details.

Parameter	Value
Shared	
Optimizer	Adam
Learning rate	$3e^{-4}$
Discount (γ)	0.99
Architecture of policy and value networks	(128, 128)
Nonlinearity	Tanh
Batch size	2048
L2 regularization	0.001
DQN/DQN-PE	
Size of replay buffer	10000
Epsilon for exploration	0.9
Epsilon decay interval	50
Epsilon decay method	exponential
Epsilon decay ending	0.05
TRPO	
Maximum KL divergence	0.01
Damping	0.01
MAML	
Meta optimizer	TRPO

CHAPTER 4

Explanation in Communicative Learning

In the past decade, machine learning has tackled many problems with noisy real-world inputs with impressive performance, fueled by large datasets. Much of this advancement has been due to uninterpretable, black-box systems. Meanwhile, the community has realized the necessity of machine interpretability [ZZ18, ZNZ18] for safety-critical applications. Intrinsically, most of the existing models are not designed to simultaneously maximize both the performance and explainability [GA19], resulting in a need for a trade-off between the performance and explainability. This trade-off often leads to a debate between the black-box models *vs* the white-box models: Models with high performance usually lack explainability, whereas models with relatively high explainability often perform poorly in real-world scenarios.

Recent trends in neural-symbolic approaches [YWG18, MGK18, LHH20, PMS16] refute the above need for the trade-off; a hybrid model could possess high performance in complex reasoning tasks while maintaining relatively high interpretability. Significantly, a robot system presented in Chapter 2 has recently demonstrated the efficacy of such an approach using a large-scale, between-subject study [EGL19]. The finding echoes the above conclusion: Forms of explanation that are best suited to foster trust do not necessarily correspond to the model components contributing to the best task performance; by integrating model components to enhance both task execution and human trust, a machine system could achieve both high task performance and high human trust. Crucially, it also shows that the means of delivering explanations matters: Providing high-level summaries is not sufficient to foster

human trust. Such explanations should not be decoupled from the participants' observations of the robot's task execution.

Despite the above progress, existing systems demonstrating specific levels of explanations are still rudimentary in terms of the forms of explanations. Existing systems mostly emphasize *hierarchical decompositions* (either spatial or temporal) of the systems' inner decision-making process, either by visualizing the saliency/attention maps of deep neural network's layers [ZNZ18, ZZ18, AWZ20, ZWW20], or by tracing top-down/bottom-up process of the graph/tree structures [LZS18, EGL19, EQZ19, EMQ20, ZRH20, ZZZ20]. Thus, the explanations and interpretability are primarily *machine-centric*; the process only unfolds the model for a human user to probe or inspect. Critically, human users' active interactions or inputs with the systems rarely change the behavior of the machine's decision-making process, and the machine's responses are primarily based on pre-computed and stored information. We call this the *passive machine—active human* paradigm, wherein an active human user may *query* the state of the machine to *passively* acquire explainable information.

We argue that human-machine teaming should follow a different and more user-friendly paradigm, which we call the *active machine–active human* [QLZ20] paradigm. In such a new paradigm, the machine would adopt the human user's input and change its behavior in *real-time* so that the system and the human user would *cooperatively* achieve a common task. Hence, such a cooperation-oriented human-machine teaming would require the machine to possess a certain level of theory of mind (ToM): A machine would behave like a human agent to *actively* infer the human user's belief, desire, and goals [YLF20, GGZ20]. The system's design is no longer limited to display its decision-making process, but further to understand human's needs to cooperate, therefore forming a *human-centric* process. Critically, the essence to establish such a cooperation lies in the *shared agency* [TSZ20, SZZ20] or *common mind* [Tom10].

Motivated to build an XAI system with the aforementioned characteristics capable of understanding human user's beliefs, design, and goals, we move from conventional explanation

tasks on function approximation (*e.g.*, classification) to tasks involving sequential decision-making. These decision-making tasks include extensive human-machine teaming, dealing with complex constraints over problems intractable to the human's inferential capabilities. By resolving the discrepancy between robot and human expectations and mental models, we hope the XAI system will assist the human user to discover the provenance of various artifacts of a system's decision-making process over long-term interactions even as the physical world evolves [GA19, CSK20]. We believe this research direction is the prerequisite for generic human-machine teaming.

The work presented in this chapter was completed in collaboration with Luyao Yuan, Xiaofeng Gao, Zilong Zhen, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. The authors' contributions include developing the simulation environment, designing and running human experiments, and data analysis. All other portions of the project were not completed by the author.

4.1 Scout Exploration Task

We devise a human-machine teaming system presented as a collaborative game, in which the human user needs to work together with a group of robot scouts to accomplish some tasks and optimize the group gain. In this game, the human user and robot scouts communicate on a constrained channel: Only the robot team directly interacts with the physical world; the human user does not directly access the physical world or direct control over robot scouts' behavior. Meanwhile, only the human user has access to the ground-truth value function that encodes human preferences about how the task should be completed (*e.g.*, minimize overall time); the robot team has to infer this value function through human-machine teaming. Such a setting realistically mimics real-world human-machine teaming tasks, as many systems perform autonomously in dangerous settings under human users' supervision where preferences are challenging, if not impossible to encode.

The XAI system is expected to provide appropriate explanations to justify its behaviors and gain human user's trust and reliance. This process is achieved by actively inferring the human user's mental model (*i.e.*, value and utility as the instantiation of the belief, desire, and goals) during the game. Therefore, the system's explanation generation is a *bidirectional* dialogue framework: The XAI system needs to both "speak" and "listen"—explaining what it has done and plans to do based on its inference of the human user's value and utility. In the meantime, the human user is tasked to command robot scouts to reach the destination while maximizing the team's score. Hence, the human user's evaluation of the XAI system is also a *bidirectional* process: The human user has to infer the goal of robot scouts and check if it aligns with the given value function of the task. Ultimately, if the XAI system works well, the robot scout value function should align well with the ground-truth value function given only to the human user, and the human user should gain high trust from the XAI system. Our methodology studies XAI in a full-blown communication system, a combination of theory-of-mind, communicative learning, value-alignment, and causal reasoning for effective explanation generation.

Our design encourages natural human-machine teaming and bidirectional reasoning as both parties have crucial but private information at the beginning of the game. The robot scouts possess information about the map but lack access to the human user's value function, which determines mission goals, hindering the robot scouts' ability to make proper decisions that reflect the human user's intent. Meanwhile, the human user, who knows the task's value function that governs the decision-making process, lacks direct access to the environment. By allowing constrained communication to fulfill human-machine collaboration, the robot scouts can make sporadic action proposals to the human user, and the human user provides a binary accept or reject feedback, and the robot scouts use that feedback to infer the human user's value function and adjust their behaviors accordingly. Based on adjusted behavior, the human user will provide ratings for the trust and reliance of the XAI system. In our setting, the communication's main purpose is to align the value function between the human

user and the robot scouts. For a fast alignment, the robot scouts need to know when and how to make proposals, such that feedback from the user is most informative to estimate the value function correctly. To obtain instructive feedback from the human user, the robot scouts must establish a shared agency or common mind—what the human user knows and believes, what the human user intends to do, and what are aligned and misaligned. Only based on this shared agency could the robot scouts provide explanations that properly justify previous actions and current proposals.

Besides the value alignment process, our design also involves estimation of human user's utility, *i.e.*, the human user's preference of the forms of explanations. In contrast to the objective value function given to the human user, this utility-driven human user's preference is subjective and more likely to be individually different. We argue that a properly modeling of such an individual difference plays a crucial role in gaining human trust and reliance. The human user's value function and utility together form the human user's mental state.

Our collaborative game, Robot Scout Exploration Game, has a minimal design and involves one human commander and three robot scouts. The game's objective is to find a safe path on an unknown map from the base (located at the bottom right corner of the map) to the destination (located at the upper left corner of the map). The map is represented as a partially observed 20×20 tile board, with each tile potentially holding one of the various devices and remain unobserved until a robot scout moves close enough to observe (reveal) the tile's contents.

We define a set of goals for the robot scouts to pursue as they find the path to reach the destination, including (i) saving time used to reach the destination, (ii) investigating suspicious devices on the map, and (iii) exploring tiles, and (iv) collecting resources. The game's performance is measured by the accomplishment of these goals by the robot scouts and their relative importance (weights), defined as the human user's value function. Again, this value function is only revealed to the human user, not the robot scouts.

One comparable but different setting to our human-machine teaming framework is the

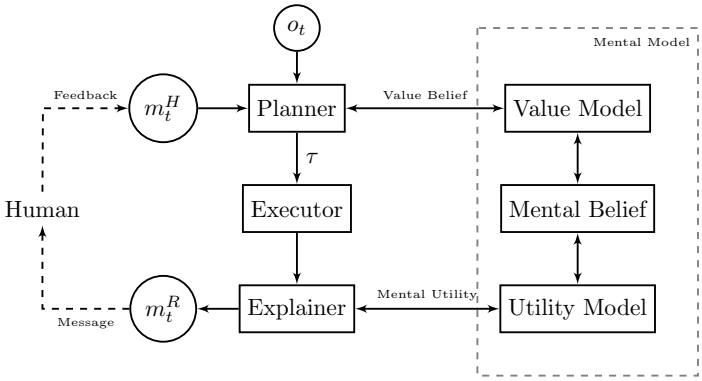


Figure 4.1: Algorithmic flow of the computational model.

inverse RL [AD21]. Inverse RL aims to recover an underlying reward function given *pre-recorded* expert demonstrations in a *passive* learning setting. In contrast, the agent (the collective form of all robot scouts) in our system is designed to learn *interactively* from *scarce* supervisions given by the human user. Crucially, our design requires the agent to actively infer the human user's mental model (value and utility) to accomplish a task *cooperatively*, a unique proper of *human-centric* learning scheme. In a nutshell, the agent is tasked to perform value-alignment by inferring the human user's mental model, actively make proposals, and evaluate the human user's feedback, requiring complex and recursive mind modeling of the human user.

4.2 Communicative Learning with Theory-of Mind

In this section, we provide an overview of the game flow and the corresponding computational model. Throughout the chapter, we use R and H to denote the robot scouts and the human user, respectively. θ encodes the parameters of the value function, s is the physical state, $b(\cdot)$ is the belief over latent variables, $x = (b_s, b_\theta, b_\upsilon)$ is the mental state (value and utility) of the human user, and m is the message used for human-machine communication. BU stands for the belief update sub-processes, where BU_1 is on the physical state, and

Algorithm 1: High-level game flow.

Set $t = 1$, initialize s^t, agent's mental state x_0^R;
while *stop condition is not satisfied* **do**
  $o_t \sim O(s_t)$ // collect observation from the environment
  $\widehat{x_t^R} = BU_1(x_{t-1}^R, o_t)$ // update belief given observation
  $m_t^R \sim \lambda_R(\widehat{x_t^R})$ // generate message (proposal & explanation) to the user
  $x_t^R = BU_2(\widehat{x_{t-1}^R}, m_t^R, m_t^H)$ // update belief given user feedback
  $\mathbf{a}_t^R \sim \pi(x_t^R)$ // agent's policy
  $s_{t+1} \sim T(s_t, \mathbf{a}_t^R)$ // state transition
  $t = t + 1$
end

BU_2 is on the value function. λ_R manages the generation of the messages to the user, including proposal and various modes of explanations. Other notations (o, t, O, T, and π) follow standard partially observable Markov decision process (POMDP) [SV10] definitions; see Table 4.1 for a summary of the notations.

Every round of the game starts with the robot scouts receiving observations from the environment and making a task plan based on their current mental state. Next, they send messages (proposals and/or explanations) to the human commander for feedback. The feedback is used to make final movement plans to execute for this round and then the scouts execute the plans. A high-level game flow is sketched in Algorithm 1, and the computation pipeline for one round of human-machine teaming is shown in Fig. 4.1. Because the game directly displays the most probable map information to the human user, we assume the communication from the agent to the human user is noise-free. After laying out the formulation of the agent policy (see Section 4.2.1), we focus on how the agent updates belief over human user's value function (BU_2) (see Section 4.2.2) and how the communication messages are

generated (λ_R) (see Section 4.2.3).

4.2.1 Agent Policy

Suppose the robot scouts already know about the human user's value function, the game simplifies to a POMDP setting, solvable by planning-based methods [SV10]. Let τ_i denote the plan proposed by the i-th scout and $\tau = \{\tau_1, ..., \tau_K\}$ as the complete plan of the scout group, where K is the number of scouts in the group. When constructing a plan, the scouts

Table 4.1: Notation used in the computational model.

Notation	Description	Remark	Notation	Description	Remark
$s \in S$	Physical State	N/A	$m^E \in M^E$	Robot's explanation	N/A
$o \in O$	Observation	N/A	$m^P \in M^P$	Robot's proposal	$\mathcal{T} \subset M^P$
$t \in T$	Time Step	N/A	$m^R \in M^R$	Robot's message	$m^R = (m^P, m^E)$
$\theta \in \Theta$	Human's value function	N/A	$fb \in FB$	Proposal feedback	$m^H(fb) \in \{0,1\}^K$
$\upsilon \in \Upsilon$	Human's utility function	N/A	$ss \in SS$	Satisfactory Score	$SS \subset \mathbb{Z}^+$
$\mathbf{a}^R$	Joint action of all scouts	$\mathbf{a}^R = (a_1^R, ..., a_K^R)$	$m^H \in M^H$	Human's message	$m^H = (fb, ss)$
b	Belief over hidden variables	$b(\cdot)$ means the belief function	λ_R	Robot's communication policy	$X^R \times M^R \longrightarrow [0,1]$
$x^R \in X^R$	Robot's mental state	$x^R = (b(s), b(\theta), b(\upsilon))$			
T	Physical State Transition Model	$S \times \mathbf{A}^R \times S \longrightarrow [0,1]$			
π	Agent Policy	$X^R \times \mathbf{A}^R \longrightarrow [0,1]$			
$\tau \in \mathcal{T}$	Group motion plan	$\mathcal{T} = (\mathbf{A}^R \times O)^*$			
		τ_i means the $i-$th scout's plan. $\tau_i \in \tau$.			

utilize the following policy:

$$\begin{aligned}\tau^* = \arg\max_{\tau} \mathop{\mathbb{E}}_{s\sim b(s),\theta\sim b(\theta)}[\theta^T f(\tau, s)] &= \arg\max_{\tau} \mathop{\mathbb{E}}_{s\sim b(s)}[f(\tau, s)]^T \mathop{\mathbb{E}}_{\theta\sim b(\theta)}[\theta] \\ &\approx \arg\max_{\tau} \bar{\theta}^T\big(\frac{1}{N_S}\sum_{n=1}^{N_S} f(\tau, s_n)\big) = \arg\max_{\tau} \bar{\theta}^T\overline{f(\tau)},\end{aligned} \tag{4.1}$$

where $f(\tau, s)$ is the fluent [NC36] when the game terminates given the state s and the scouts' plan τ, and the above equation takes the hard-max for plan selection. Given the dynamics of the game, f can be forward simulated in our planner, such that the expectation of $f(\tau, s)$ can be approximated using Monte Carlo methods with state samples. Instead of computing the full distribution, the agent only needs to keep track of the mean of the belief over human user's value function as we are using a linear model to calculate the gain of the game; we use $\bar{\theta}$ to denote the mean of $b(\theta)$. We can use the Boltzman rationality model to convert the planning problem described in Eq. (4.1) to a stochastic process, *i.e.*:

$$p(\tau; \bar{\theta}) = \frac{\exp\left(\beta_1 \bar{\theta}^T \overline{f(\tau)}\right)}{\sum_{\tau'\in\mathcal{T}} \exp\left(\beta_1 \bar{\theta}^T \overline{f(\tau')}\right)}, \tag{4.2}$$

where $\beta_1 \geq 0$. This conversion facilitates the inference of the human user's value function by enabling gradient-based optimization methods to learn $\bar{\theta}$. After a plan τ is determined, the joint action of all robot scouts is the first action of the plan, $\mathbf{a}^R = (\tau_1[0], \ldots, \tau_K[0])$.

4.2.2 Value Function Estimation by Modeling ToM

The human user's value function is unknown to the scouts and has to be learned through interaction, raising challenges for classic POMDP solvers. To estimate the human user's value function during the communication process, we integrated ToM into our computation model and developed a closed-form learning algorithm. Our algorithm leverages the assumption that, given a cooperative human user, the accepted plans are more likely to have a performance advantage over the rejected ones.

4.2.2.1 Belief Update with Level-1 ToM

We use $m^H(fb)$ to denote the human user's feedback, which is a binary code with the i-th bit indicating the acceptance or rejection of the proposal from the i-th scout. Assuming the human user is following the above decision-making process, the likelihood function of human user's feedback is:

$$p(m^H(fb)|\tau;\bar{\theta}) = \prod_{i=1}^{K} p(\tau_i;\bar{\theta})^{m^H(fb)_i}(1-p(\tau_i;\bar{\theta}))^{(1-m^H(fb)_i)}, \tag{4.3}$$

where $p(\tau_i;\bar{\theta}) = \sum_{\tau\in\mathcal{T},\tau_i\in\tau} p(\tau;\bar{\theta})$. Given this likelihood function, we can learn the mean of the parameter of value function $\bar{\theta}$, following the maximum likelihood estimation (MLE) derivation by maximizing $\log p(m^H(fb)|\tau;\bar{\theta})$ w.r.t. $\bar{\theta}$. Because $\bar{\theta} > 0$ and $\|\bar{\theta}\|_1 = 1$, this MLE process can be calculated by the projected stochastic gradient ascent algorithm [Nes03], yielding a closed-form derivation for $\frac{\partial \log p(m^H(fb)|\hat{\tau};\bar{\theta})}{\partial\bar{\theta}}$:

$$\begin{aligned}\frac{\partial \log p(m^H(fb)|\hat{\tau};\bar{\theta})}{\partial\bar{\theta}} = \beta_1 \sum_{i=1}^{K} \Big[\mathbf{1}(m^H(fb)_i = 1)\Big(\sum_{\tau\in\mathcal{T}\hat{\tau}_i\in\tau} \frac{\exp\left(\beta_1\bar{\theta}^T\overline{f(\tau)}\right)}{\sum_{\tau'\in\mathcal{T},\hat{\tau}_i\in\tau'}\exp\left(\beta_1\bar{\theta}^T\overline{f(\tau')}\right)}\overline{f(\tau)}\Big) + \\ \mathbf{1}(m^H(fb)_i = 0)\Big(\sum_{\tau\in\mathcal{T}\hat{\tau}_i\notin\tau} \frac{\exp\left(\beta_1\bar{\theta}^T\overline{f(\tau)}\right)}{\sum_{\tau'\in\mathcal{T},\hat{\tau}_i\notin\tau'}\exp\left(\beta_1\bar{\theta}^T\overline{f(\tau')}\right)}\overline{f(\tau)}\Big) - \mathop{\mathbb{E}}_{\tau\sim p(\tau;\bar{\theta})}[\overline{f(\tau)}]\Big],\end{aligned} \tag{4.4}$$

where the two indicator functions select which summation to take conditioned on the feedback of the $i-$th proposal. The summation over weighted fluents, despite the overwhelming form, can be interpreted as the expected fluents in accord to the accepted/rejected plans. The intuition of this gradient is the difference between the expected fluents from plans without the accept/rejected proposals and the expected fluents from all the plans.

4.2.2.2 Belief Update with Level-2 ToM

The above belief update mechanism assumes the human user will provide feedback to the proposals based on the intrinsic value of the proposals, *i.e.*, the expected return of the proposed plans given the underlying parameters of the value function. However, this is

unlikely to be the case, as completely rational agents do not exist. Thus, we need to properly model level-2 ToM: With the explanation generated by the XAI system (see Section 4.2.3 for details), we further assume that the human user will be cooperative and provide feedback to best accelerate the parameter learning. Suppose the human user provides feedback based on the improvement brought by the feedback, we have

$$q(m^H(fb)|\theta^*, \bar{\theta}, \tau) = \frac{\exp\left(-\beta_2 \|\bar{\theta} + \eta_t \frac{\partial \log p(m^H(fb)|\tau;\bar{\theta})}{\partial \bar{\theta}} - \theta^*\|^2\right)}{\sum_{\widehat{m^H(fb)} \in FB} \exp\left(-\beta_2 \|\bar{\theta} + \eta_t \frac{\partial \log p(\widehat{m^H(fb)}|\tau;\bar{\theta})}{\partial \bar{\theta}} - \theta^*\|^2\right)}, \tag{4.5}$$

where $\beta_2 \geq 0$ controls the extremeness of the softmin, η_t is the learning rate at time t, and θ^* is the ground-truth parameters of the value function possessed by the human user. The intuition of this equation is: The feedback from the human user is sampled from a softmin distribution of the distance between the updated parameters given the feedback and the ground-truth parameters. The smaller the distance is, the larger the improvement brought by that feedback, and the larger the improvement is, the more likely the feedback is provided. Further analysis of the above distance can be found in [LDL18]. Here, we use a softmin instead of hardmin in the data selection process. Integrating this feedback function into our parameter learning algorithm, we can derive a new parameter update function:

$$\bar{\theta}^{t+1} = \bar{\theta}^t + \eta_t g\big(m^H(fb)\big) + 2\beta_2 \eta_t^2 \Big(g\big(m^H(fb)\big) - \mathop{\mathbb{E}}_{m(fb)\sim q(\theta^*, \bar{\theta}^t, \tau)} \big[g\big(m(fb)\big)\big]\Big), \tag{4.6}$$

where $g\big(m(fb)\big) = \frac{\log p(m(fb)|\tau;\bar{\theta}^t)}{\partial \theta}$. The first two terms are the same as the level-1 belief update, whereas the third term grasps the message's context by comparing the selected message against the also-runs and leverages the advantage to further update the belief. Notice that θ^* is unknown to the agent, so q in the expectation dose not have an exact solution. Thus, we use $\bar{\theta}^t + \eta_t g\big(m^H(fb)\big)$ as an approximation of θ^*. That is, we first calculate level-1 ToM update on the parameters of the value function, then we take an additional gradient ascent step for level-2 ToM update.

4.2.2.3 Proposal Generation

The XAI system generates proposals in accord to the change of expected belief. At each step, the agent first computes a new $\bar{\theta}'_m$ for each $m \in M^H$. Next, the change of expected belief can be calculated by $\delta(\tau, \bar{\theta}) = \mathbb{E}_{m \sim p(m^H|\tau,\bar{\theta})}[\|\bar{\theta}'_m - \bar{\theta}\|_2]$ for each $\tau \in \mathcal{T}$. If $\max_{\tau \in \mathcal{T}} \delta(\tau, \bar{\theta})$ surpasses a given threshold, the robot scouts make a proposal with $\arg\max_{\tau} \delta(\tau, \bar{\theta})$. This formulation is generic; we can also substitute in other measurement (*e.g.*, the expected variance of the $\bar{\theta}'$) in terms of the change of expected belief to generate more diverse update.

4.2.3 Explanation Generation by Modeling Mental Utility

We generate explanations alongside proposals to aid the human user to make decisions. Given trajectories produced by the planner, the explainer aims to generate human-like explanations that not only provide sufficient information but also match the human user's language preferences, *i.e.*, the mental utility.

Formally, an explanation is defined by its semantic inputs and a set of syntactic rules. The former is to provide explanations regarding *what*, including the current observation o, physical state s, and belief over the value function $b(\theta)$. The latter is to provide explanations regarding *how*. The explainer model is to determine the optimal syntax that matches the human user's mental utility. Specifically, we predefine a set of attributed templates; these templates provide the basis of an explanation and are filled in according to relevant attributes. At each step, the explainer predicts the human user's most favorable attributes based on the satisfactory score. We propose a sequential explanation generation model capable of adopting the temporal dynamics of the human's mental state; it defines utility functions to synthesize the most efficient and suitable explanations.

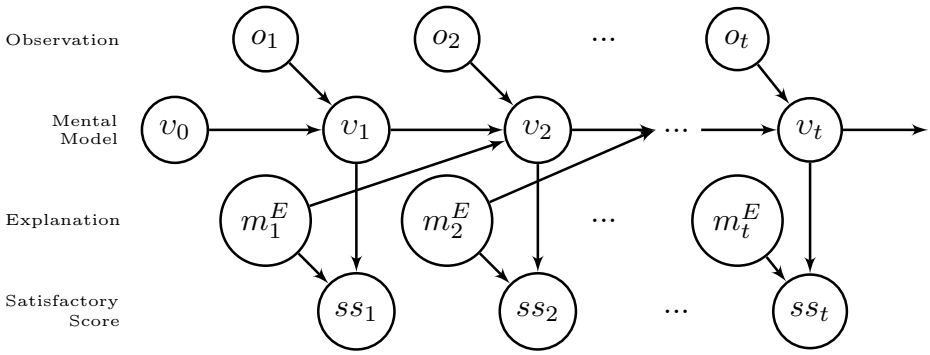


Figure 4.2: Temporal evolution of explanation generation as a function of t.

4.2.3.1 Sequential Explanation Generation

At time step t, the explainer takes in a tuple $h_t = \{(m_{t-1}^E, ss_{t-1}, o_t)\}$ as input, where $m_{t-1}^E \in M^E$ is the explanation of previous round, $ss_{t-1} \in SS$ is user's satisfactory score estimated by human user's feedback of the previous round, and $o_t \in O$ is the current observation. Given the sequential input history $H_t = \{h_k, k = 1, ..., t\}$, the explanation objective is to generate an explanation m_t^E that maximizes the expected score:

$$m_t^E = \arg\max_{m^E \in M^E} \mathbb{E}_{\hat{ss} \sim p(ss|H_t)}[\hat{ss}(a^E)] - \lambda_c \text{cost}(m^E), \tag{4.7}$$

where $a^E \in A^E$ is an extracted attribute vector of m^E, cost$(\cdot)$ a pre-defined cost function, and λ_c a constant factor.

We model the process of computing $\mathbb{E}[\hat{ss}(a^E)]$ as a hidden Markov model (HMM) by introducing a mental state variable $\upsilon \in \Upsilon$, which corresponds to the human user's mental utility of the explanation; see Fig. 4.2 for the graphical illustration of the computing process. At time step t, we compute the expected score as:

$$\begin{aligned} \mathbb{E}_{\hat{ss} \sim p(ss|H_t)}[\hat{ss}(a^E)] &= \sum_{ss \in SS} p(ss|a^E, ss_{1:t-1}, o_{1:t}, a_{1:t-1}^E) ss \\ &= \sum_{ss \in SS} \left(\sum_{\upsilon_t \in \upsilon} p(\upsilon_t|ss_{1:t-1}, a_{1:t-1}^E, o_{1:t}) p(ss|\upsilon_t, a^E) \right) ss. \end{aligned} \tag{4.8}$$

Algorithm 2: Explanation Generation

Input : *templates* - all explanation templates

Output: $\{m_1^E, m_2^E, ...\}$

$t \leftarrow 1$

while *not stopped* **do**

  explanations $\leftarrow$ FillSlots(*templates*)

  Get O_t, ss_{t-1} from agent

  $m^E \leftarrow None$

  for m_i^E **in** *explanations* **do**

    $a^E \leftarrow$ ExtractAttribute(m_i^E)

    Compute $\mathbb{E}[\hat{ss}(a^E)]$ according to Eq. (4.11).

    $m^E \leftarrow \arg\max_{\{m^E, m_i^E\}} \mathbb{E}[\hat{ss}(a^E)] - \text{cost}(m^E)$

  end

  $m_t^E \leftarrow m^E$, $t \leftarrow t+1$

end

Let $\mathcal{K}(a_{t-1}^E, o_t) = p(\upsilon_t|\upsilon_{t-1}, a_{t-1}^E, o_t)$ be the transition matrix that encodes the transition probability from mental states υ_{t-1} to υ_t, and $\mathcal{F}(a^E) = p(ss|\upsilon_t, a^E)$ be the score function that models the distribution of satisfaction scores. We have:

$$p(\upsilon_t|ss_{1:t-1}, e_{1:t-1}, o_{1:t}) = \sum_{\upsilon_{t-1}\in\upsilon} p(\upsilon_{t-1}|ss_{1:t-1}, a_{1:t-1}^E, o_{1:t-1})\mathcal{K}(a_{t-1}^E, o_t), \tag{4.9}$$

where $p(\upsilon_t|ss_{1:t}, a_{1:t}^E, o_{1:t}) = \alpha_t$ is computed by an iterative process:

$$\begin{aligned} p(\upsilon_t|ss_{1:t}, a_{1:t}^E, o_{1:t}) &\propto \mathcal{F}(a_t^E) \odot \left(\mathcal{K}(a_{t-1}^E, o_t)^T p(\upsilon_{t-1}|ss_{1:t-1}, a_{1:t-1}^E, o_{1:t-1})\right) \\ &= \mathcal{F}(a_t^E) \odot \left(\mathcal{K}(a_{t-1}^E, o_t)^T \alpha_{t-1}\right), \end{aligned} \tag{4.10}$$

where $\odot$ is an element-wise product operator. Therefore, Eq. (4.8) can be written as

$$\mathbb{E}_{\hat{ss}\sim p(ss|H_t)}[\hat{ss}(a^E)] = \sum_{ss\in SS} \frac{ss}{Z} \alpha_t^T \mathcal{K}(a_{t-1}^E, o_t)\mathcal{F}(e), \tag{4.11}$$

where Z is a normalization constant of $p(ss|H_t)$; see Algorithm 2 for the computational flow.

4.2.4 Explanation with Ontogenetic Ritualization

Literature in evolutionary anthropology demonstrates strong evidence that early infants learn to communicate, especially in a symbolic manner, not based on imitation but rather on an individual learning process termed *ontogenetic ritualization* [MN12, Tom10, Loc80]. [TC97] argue such communicative behavior as a communicative signal that can be formed by two individuals shaping each other's behavior in repeated instances of interaction over time. Similar phenomena have also been observed and investigated on other primates, such as great apes [HRT13, Tom96]. For example, many individual chimpanzees come to use a stylized "arm-raise" to indicate that they are about to hit the other and thus initiate play [TC97]. In this way, a behavior that was not at first a communicative signal would become one over time. Generally, we follow [TZ02] to define the process of ontogenetic ritualization: (i) individual A performs behavior X; (ii) individual B reacts consistently with behavior Y; (iii) based on the initial steps of X, B anticipates A's performance of X, and hence performs Y; and finally, (iv) A anticipates B's anticipation of X, and hence produces X in ritualized form so as to elicit Y.

We argue that the process of ontogenetic ritualization can also be formed during human-robot teaming, specifically when understanding and reacting to explanations. To achieve this goal, we set the "ritualized form" as a subset of explanation attributes A^E. As such, the computational model described here allows robot scouts to generate ritualized explanation based on their anticipation of human feedback, *i.e.* $\mathbb{E}[\hat{ss}(a^E)]$.

4.3 Human Subject Experiments

4.3.1 Participants Description

Participants for this study were recruited from the online Prolific user research platform. Participants were selected based on their location (in the United States), their highest level

of education (at least a bachelor's degree), and the device they were using (no mobile users were selected). This choice was made to confine our participants to a population that is more likely to understand the nuance of the game while maintaining a broad pool of participants who are representative of the general population. A desktop/laptop computer was required to interact with the game appropriately. Information on the participant's computer was collected (*i.e.* User-Agent). No other demographic information from the participant was collected after passing our demographic selection criteria.

After participants finished the introductory material, a 7-question familiarity test was given to participants before proceeding into the game. This check was to make sure participants understood the rules of the game, what their objectives were, how to interpret value functions and the distinction between explanations and proposals. Participants passed the questionnaire if they answered every question correctly. If a participant missed a question, a page was shown to explain the correct answer. Participants who missed a question had to repeat the entire questionnaire, and participants who failed to pass the questionnaire twice were removed from the study.

Participants were assigned randomly to each condition and were balanced automatically by our survey platform (Qualtrics). Compensation started at $10 USD per participant, and our scoring system incentivized participants to score as many points as possible. Participants received $0.05 USD per point in the game, with a maximum total payout of $20 USD per participant.

4.3.2 Study Design

The study was conducted in a between-subject design. Participants were randomized in a hierarchical group selection process: an outer hierarchy and an inner hierarchy; see Fig. 4.3. The outer hierarchy was randomized to assign participants evenly based on mental model questions: (i) value function and (ii) behavior prediction. The inner hierarchy was randomized to evenly assign participants based on different explanation formats: (i) a proposal

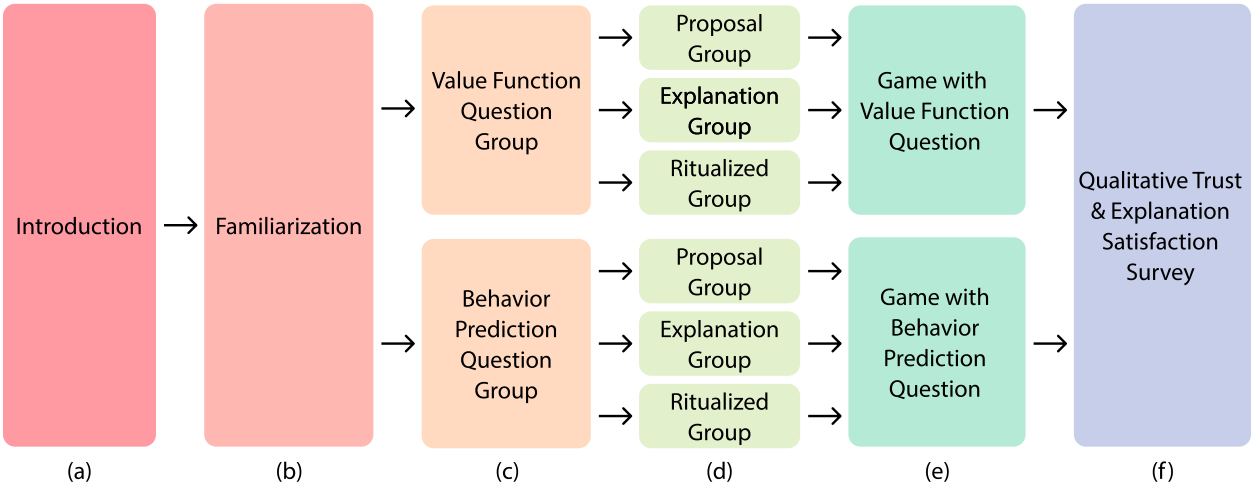


Figure 4.3: User study flow. (a) Participants begin with an introduction to explain the setting and define key terms. (b) Participants are then familiarized with the game interfaces, and a questionnaire is given to verify participants understand the game. Participants that did not pass the familiarization were removed from the study. (c) Participants are randomly split into two groups: a group that is asked to infer the robot scout's current value function and a group that is asked to predict the robot scout's next behavior. This is done in a between-subject design. (d) Participants are further randomly split to receive different forms of explanations: proposals, explanations, and ritualized explanations. This is done in a between-subject design. (e) The participants then play the game and are asked the question assigned to their group throughout the experiment. (f) After finishing the game, participants were asked qualitative trust and explanation satisfaction questions.

group, (ii) an explanation group, and (iii) a ritualized explanation group. Among three groups, the robot scouts will follow the exact same action policy, π, and belief update process, BU. The groups differ only by the explanations forms received by the human participant, λ_R, and the question about the robot scouts' plan (current *vs* next round).

Our study includes four variables. The only independent variable is the form of the explanation a participant received: proposal, explanation, or ritualized explanation. Three dependent variables are (i) value function alignment, (ii) behavior prediction, and (iii) qual-

itative trust and explanation satisfaction. To ensure no confounding on order or effects from answering a question regarding value function prediction and behavior prediction, the study was designed with the above outer hierarchy using a between-subject design. Participants from all groups were asked to provide qualitative trust and explanation satisfaction responses throughout the game.

The general study procedure was identical for all subject groups. Participants were first introduced with background information on tasks. Specifically, the participants were tasked to behave as a commander to guide a group of robot scouts to move from bottom right to upper left on the map while collecting resources. They were informed about the information asymmetry: They know about the group-truth value function, but they have no direct access to the environment; only the robot scouts can directly interact with the environment but have no access to the value function. Participants were informed to work cooperatively with the scouts to achieve a high score, and they would be compensated more with a high score. Next, participants were presented with the game interface, with a focus on functions and operations on various panels. Depending on their group, participants received instructions that only cover the panels presented in their group. By the end of the introduction, participants were challenged with questions. Participants who did not answer these questions correctly were removed from the study.

In Fig. 4.3, we outline the flow of the experiment to give a high-level overview of the participant's experience through the game. The figure shows our between-subject design across our two mental model questions (value function and behavior prediction) and our explanation formats (proposal, explanation, and ritualized).

The introduction phase of the experiment introduces the basic background of the game. The introduction outlines that the participants are commanders in charge of finding a path from the lower-right-hand corner of the map to the upper-left-hand corner of the map. The introduction outlines that the area may have dangerous devices, such as bombs, along the path. Participants are told they have a team of robot scouts to help explore the area, and

that the scouts will provide proposals (and in explanation groups, explanations).

During familiarization, participants are instructed that while their objective is to get to the upper-left corner, they are also instructed there are sub-goals the team would benefit from achieving. These sub-goals consist of time, area explored, number of bombs investigated, and resources collected. Participants are then informed that these goals are specific to the team's current circumstance and the value function conveys the relative importance of these sub-goals. Participants are informed they will be scored, and this score is weighted by the value function. Finally, the robot scout proposals and explanations are described. Proposals correspond to robot scout plans, and explanations attempt to justify those plans. These various components are introduced using figures of the panels shown in Fig. 4.4.

To test the participant's understanding of the background, the participant is given several attention check questions at the end of familiarization. For example, we asked "for each trial, you will be given a value function that describes the team's current mission priorities" (correct answer: true). Participants who answered a question incorrectly would receive instruction as to why the correct answer is correct and would be required to repeat the attention check. Participants who failed the attention check twice did not further participate in the study.

We implemented this game using HaxeFlixel, a 2D Game Engine used to create JavaScript-based games. Participants can access the game on web browsers. The full user interface of the game is displayed in Fig. 4.4. Throughout the study, the participant monitors the progress of the team, receives explanations, and gives feedback by accepting or rejecting the proposals. The team's performance is quantified as a score, which reflects how well the scouts can estimate the participant's value function and act accordingly. Participants are instructed to maximize the team's score. The score is weighted by the value function to score the relative importance of sub-goals. Each sub-goal score is computed from the environment's reward function.

During the game, the robot scouts attempt to infer the human value function. To infer

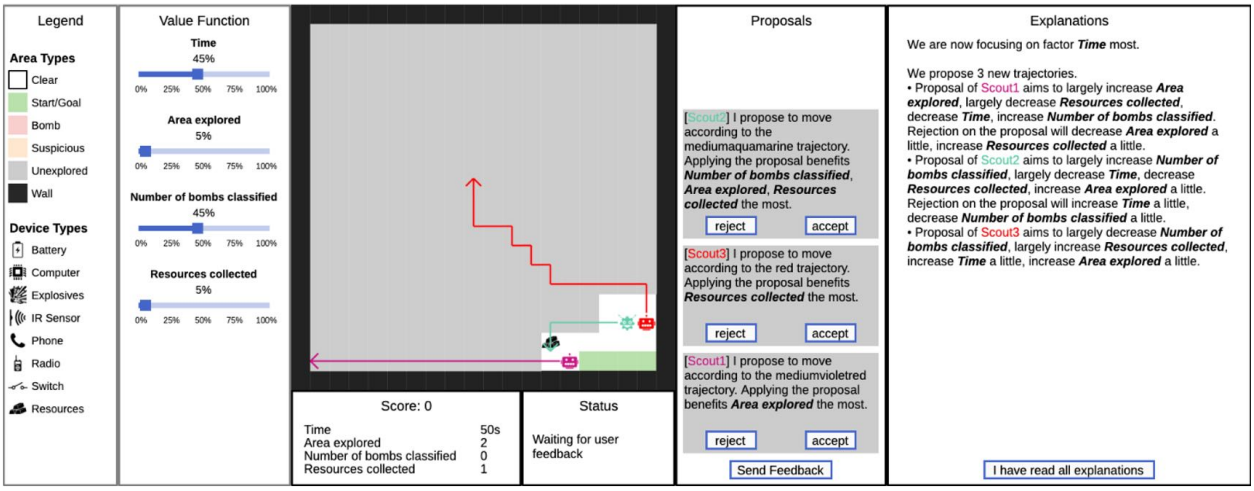
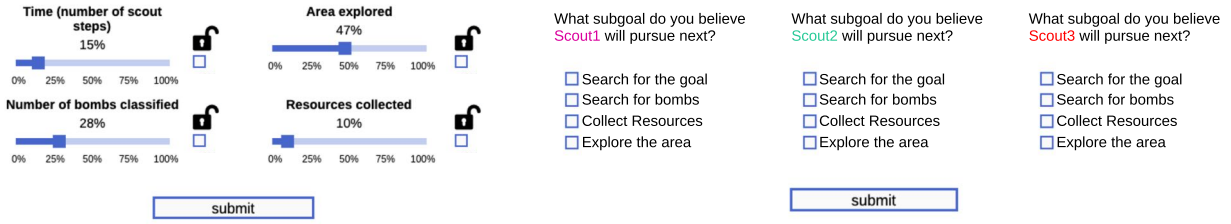


Figure 4.4: User interface of the scout exploration game. Moving from left to right, the Legend panel displays a permanent legend for the participant to refer to understand different tile types. The Value Function panel shows the value function of the participant's team, is unknown to the robot scouts, and cannot be modified by the participant. The central map shows the current information on the map. The Score panel shows the participant's current score and the individual fluent functions that contribute to the score. The overall score is calculated as the normalized, value function-weighted sum of the individual fluent function scores. The Status panel displays the current status of the system. The Proposal panel shows the robot scouts' current proposals, and the participant can accept/reject each. The Explanation panel shows explanations provided by the scouts.

the correct value function, the robot team proposes action plans to the participant and estimates the value function based on the participant's feedback. Explanations accompany the proposals to clarify the motivation of the robot scouts. An example proposal is: "We can keep moving despite the suspicious area (proposal) if we want to find a path from A to B as soon as possible (explanation)." If the participant accepts this proposal, the robots will increase the value of time in the value function. Otherwise, the robots will increase the value of investigating bombs and tile exploration but decrease the value of time. The game repeats in a loop, where robot scouts make proposals (and in some groups, explain), execute



(a) Value function question interface

(b) Behavior prediction question interface

Figure 4.5: Example interfaces for the value function question and the behavior prediction question. (a) Participants can slide the bars to set a relative importance of each sub-goal. The sub-goals must sum to 100%. As the participant changes one slider, the others will automatically decrease to keep the sum at 100%. Participants can lock a particular slider by checking the lock symbol to the right of the slider. (b) Participants are asked to predict which sub-goal the robot scouts will pursue next. Participants are asked to predict the sub-goal for each scout individually; this is because proposals are generated on a per-scout basis.

plans, and repropose until they find a path to the upper-left-hand corner of the map.

Our between-subject design is divided by the question type that will be asked during the experiment (value function or behavior prediction) or and by the explanation format displayed to the participant (proposal, explanation, or ritualized). Participants are asked the dependent measure question before the robot scouts start the next round of explanation and proposing. The value function question asks participants to provide the value function *they believe* the robot scouts are using. Participants provide their rating by manipulating a set of sliders that are interdependent; the slides must always sum to 100%. This provides the relative importance of sub-goals. For the behavior prediction question, participants are asked to predict what proposal the robot scouts will make next. Note that this is a between-subject design, so participants will see one question but not the other.

For the explanation format displayed, participants in the proposal group will see only

robot scout proposals (see Proposal Panel in Fig. 4.4), while participants in the explanation group will see robot scout proposals and explanations (see Explanation Panel in Fig. 4.4). The ritualized group is identical to the explanation group, except that robot scouts actively attempt to ritualize explanations based on the shared common mind between the robot scouts and the participant.

After the participants finish the game, they are directed to a post-experiment survey to evaluate qualitative trust and explanation satisfaction. Self-reported trust is evaluated using Likert-scale questions, which are designed based on Muir's questionnaire [Mui94] and Madsen's Human-Computer Trust Instrument [MG00]. The questionnaire intends to evaluate how the information given to the participants across different groups helps them make appropriate decisions when they are asked to give feedback on the proposals. Such appropriate reliance [LS04] is supported by a correct understanding of multiple components of the system, including the planning, value function estimation, proposal generation, feedback interpretation, and/or explanation generation, which form the basis of trust. Specifically, the trust questionnaire comprises questions that intend to evaluate the perceived reliability, technical competence, and understand-ability of the scouts with respect to these components. We ask the participants "how much would you trust the robot scouts to achieve a high score on their own, given they have the correct value function?" and "how much do you trust the robot scouts to learn the value function of another commander in another circumstance?".

Explanation satisfaction is evaluated in the aspects of *transparency, helpfulness*, and *timeliness* via Likert-scale questions to reflect the participant's belief regarding how well the explanation has helped them understand these components and make correct feedback to guide the team towards plans that are better suited to the scenario and value function given to the participants.

4.3.3 Hypotheses

The hypotheses we are testing in this experiment are related to quantitative measures for mental model alignment and qualitative measures relating to trust and explanation satisfaction. The quantitative measures for mental model alignment include: (*H1*) value function alignment, (*H2*) behavior prediction, and (*H3*) user-machine task performance. For value function alignment (*H1*), we hypothesize that groups that have access to explanations will be more accurate inferring the current robot scout value function. For the behavior prediction (*H2*), we hypothesize that groups that have access to explanations will be more accurate in predicting what the robot scouts will do next. The user-machine task performance (*H3*) will be evaluated by the score participants receive from the game.

Our qualitative measures will assess trust by asking participants whether they would trust the robot scouts to complete the task on their own, given they have the correct value function. Additionally, we will ask participants whether they trust the scout to learn a different value function with a different commander. We hypothesize that groups that have access to richer explanations will rate the qualitative trust measures higher than those without (*H4*). Our second qualitative hypothesis (*H5*) is that groups with access to richer explanations will report higher degrees of explanation satisfaction.

4.4 Results

The experiments for this chapter are ongoing, though expected results and initial results will be presented here. Our primary measures of significance will be using a student's t-test and analysis of variance (ANOVA) using the F-test. We expect to see the following significances for each hypothesis:

- *H1*: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. No significance between explanation and ritualized explanation group.

- *H2*: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. No significance between explanation and ritualized explanation group.
- *H3*: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. No significance between explanation and ritualized explanation group.
- *H4*: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. Significance between explanation and ritualized explanation group.
- *H5*: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. Significance between explanation and ritualized explanation group.

We believe for *H1* and *H2*, we will observe significance between the proposal and explanation group. This is predominately due to the richer explanations providing a deeper insight into the robot scout's reasoning process, allowing better inference of the mental state of the robot scouts. The same applies between the proposal and ritualized explanation group. We do not expect to see a significance between the explanation and ritualized explanation for these hypotheses because these two forms of explanation convey similar information in different forms (ritualized being an abridged version of the full explanation based on the common mind between the human and the robot scouts).

For *H3* and *H4*, we believe to see significance between all groups. Between the proposal and explanation group, we believe the transparency and insight provided by the explanations will improve trust and satisfaction ratings. Furthermore, between the explanation and ritualized explanation group, we predict the ritualization will further improve trust and satisfaction ratings, as the ritualization conveys a deeper understanding of the shared common mind between the human and the robot scouts.

Fig. 4.6 shows that scouts are better able to estimate the ground-truth value function

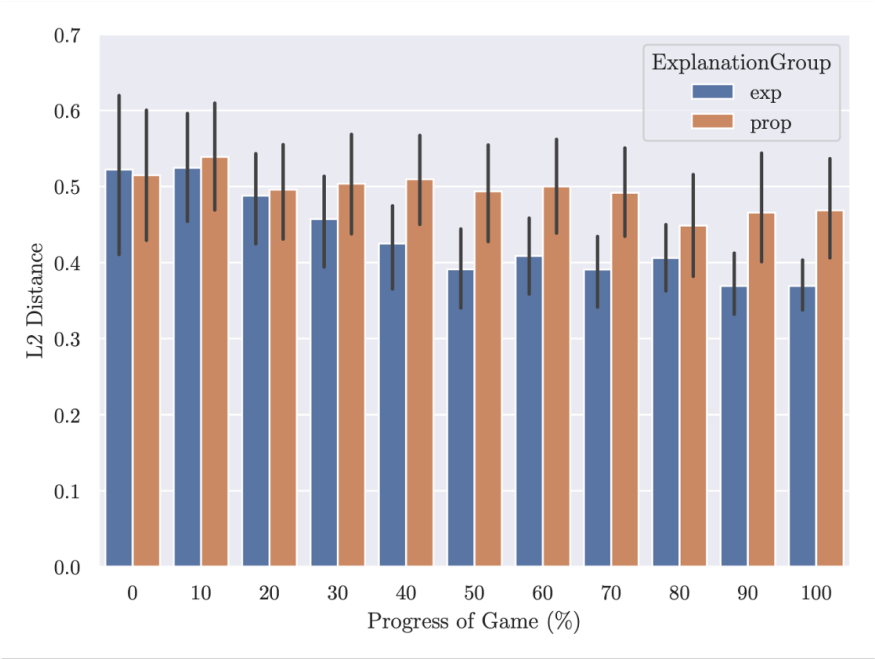


Figure 4.6: Scout value function vs. ground truth value function, measured by L2 distance between the scout value function vector and the ground truth value function vector. The explanation group (exp) achieves better performance than the proposal group (prop) as the game progresses. The percent indicates how far into the game the participant is when prompted to estimate the scout's value function. N=35.

used in the explanation group, confirming our *H1* hypothesis that value function alignment will perform better in groups that have access to explanations. At the final step (100%), we observe $t = -2.64$, $p = 0.01$, indicating a significant difference between the performance of the explanation group compared with the proposal group. Significance is also seen at 40%, 50%, 60%, 70%, and 90%.

4.5 Conclusion and Discussion

In this study, we looked at a unique XAI paradigm, namely an *active machine–active human* paradigm wherein both the machine and the human are active participants in the explanation

process. This contrasts to more traditional XAI studies that use a *passive machine—active human* paradigm wherein the machine provides an explanation that a human user interprets, with no engagement from the human back to the machine. To achieve this paradigm, we adopt a communicative learning framework based on theory of mind (ToM) where the machine actively reasons about human user's mental states. This communicative learning paradigm generates explanations that help build a *common mind* between the user and machine, thereby allowing the machine to perform the task better.

We constructed a Robot Scout Exploration Game, where a team of robot scouts explores a dangerous area, looking for a safe path for the commander's team to cross the area. The team has sub-goals, such as minimizing the amount of time or investigating devices that may be bombs. The robot scouts provide information to the commander from their sensing capabilities, along with proposals on what the scouts plan to do next and explanations for those proposals. The commander can then accept or reject the proposals, thereby providing feedback to the scouts on the utility of a proposal. The robot scouts then use this feedback to estimate the commander's intents and goals to improve future proposals and explanations. This iterative communication process continues until the team completes the task (finding a safe path to reach the destination).

The user study presented here quantitatively assesses the degree to which different forms of explanation improve mental model understanding between the user and the machine and qualitatively assess user-machine trust and explanation satisfaction. While final results are forthcoming, we expect that access to richer forms of explanation will improve the mental model understanding and user-machine task performance. Additionally, we expect richer forms of explanations to foster more trust and improve explanation satisfaction. Of note, we anticipate that these scores will be the highest in a ritualized explanation group, where the machine shortens explanations as the user and machine establish a *common mind*.

This study aims to present an unexplored *active machine–active human* paradigm where both the human and the machine actively participate in the explanation process. We be-

lieve this opens a new venue for future interactive XAI studies that showcase collaborative environments between users and machines.

CHAPTER 5

Conclusion

In this dissertation, we examined how an agent can learn generalizable knowledge from observations and interventions. The work presented here attempts to answer fundamental questions about the components necessary for generalization and explanation. We examined using a temporal And-Or graph (T-AOG), a haptic neural network, a hierarchical Bayesian model, and reinforcement learning (RL) to learn generalizable representations. We found that these models can complement each other (T-AOG and haptic neural network), perform well in generalization tasks on their own (hierarchical Bayesian model), and are not able to generalize effectively (model-free RL). We also explored concepts related to generalization and transfer, such as explainability, to examine the interplay between generalization performance and explanatory performance.

In Chapter 2, we showcased an imitation learning task where a robot learned how to open medicine bottles using two different forms of imitation: a temporal And-Or graph (T-AOG) to encode symbolic, long-term task structure and a haptic network to enable the robot to imitate the poses and forces of the human demonstrator. The two modeling components were combined using the generalized Earley parser (GEP), yielding a highly capable learner. Additionally, these model components can produce explanations, and we examined how each modeling component fostered human trust. The T-AOG contributed most to fostering human trust, but the combined GEP model performed best at the bottle opening task. This divergence shows a need to consider both task performance and explanation simultaneously to construct a capable performer and a capable explainer.

In Chapter 3, the OpenLock task tested agents' ability to form abstract causal structures and apply them to observationally different but structurally similar situations. Human subjects showed a remarkable ability to form these abstract causal structures and apply them to situations with similar but different structure. Model-free reinforcement learning (RL) was unable to form these causal structures and apply them, even under favorable training conditions. A hierarchical Bayesian learner, based on causal theory induction [GT09], showed similar learning trends as human learners. The hierarchical Bayesian learner used a top-down hypothesis space generation scheme to explore the space of possible causal structures while bottom-up, instance-level learning guiding the learner to focus on features that were likely to produce a causal effect. The combination of these two mechanisms produced a highly capable learner, suggesting that causal learning in interactive domains benefits from both structural abstraction and feature-level inductive biases.

In Chapter 4, we presented a task that requires two agents communicate to share knowledge (observations) or preferences (values). A human user collaborated with a team of robots to achieve the goal. In this setting, the need for explanation arises from each agent having partial information that must be communicated to the other agent. The communicative learning framework presented shows robot scouts capable of using feedback from a human user to properly align their internal value with the human's value and make plans in accordance with the inferred values. Value alignment is critical for the future of human-robot teaming; aligning values between humans and robots is required to foster trust and enable human-robot teaming in daily life.

Moving forward, AI and robots must be capable of explaining themselves and constructing causal representations to generalize effectively. While this dissertation made progress towards these goals, there are still many unanswered questions around what constitutes an effective explanation, how different modeling components can generate explanations, efficient causal hypotheses spaces generation in large-scale tasks, and effective inductive biases to accelerate the causal learning process.

REFERENCES

[AD21] Saurabh Arora and Prashant Doshi. "A survey of inverse reinforcement learning: Challenges, methods and progress." *Artificial Intelligence*, p. 103500, 2021.

[AWZ20] Arjun Akula, Shuai Wang, and Song-Chun Zhu. "CoCoX: Generating conceptual and counterfactual explanations via fault-lines." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[BCP16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "OpenAI Gym.", 2016.

[BDG17] Neil R Bramley, Peter Dayan, Thomas L Griffiths, and David A Lagnado. "Formalizing Neurath's ship: Approximate algorithms for online causal learning." *Psychological Review*, **124**(3):301, 2017.

[BGT18] Neil R. Bramley, Tobias Gerstenberg, Joshua B. Tenenbaum, and Todd M. Gureckis. "Intuitive experimentation in the physical world." *Cognitive Psychology*, **105**:9–38, 2018.

[BLS15] Neil R Bramley, David A Lagnado, and Maarten Speekenbrink. "Conservative forgetful scholars: How people learn causal structure through sequences of interventions." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **41**(3):708, 2015.

[Cat11] Erin Catto. "Box2d: A 2d physics engine for games.", 2011.

[CBS05] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. "Intrinsically motivated reinforcement learning." In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[CF98] Cristiano Castelfranchi and Rino Falcone. "Principles of trust for MAS: Cognitive anatomy, social importance, and quantification." In *Proceedings International Conference on Multi Agent Systems*, 1998.

[Che97] Patricia W Cheng. "From covariation to causation: a causal power theory." *Psychological Review*, **104**(2):367, 1997.

[CSK20] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. "The emerging landscape of explainable automated planning & decision making." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[DCH16] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. "Benchmarking deep reinforcement learning for continuous control." In *International Conference on Machine Learning (ICML)*, 2016.

[DL45] Karl Duncker and Lynne S Lees. "On problem-solving." *Psychological monographs*, **58**(5):i, 1945.

[Dom15] Pedro Domingos. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books, 2015.

[Ear70] Jay Earley. "An efficient context-free parsing algorithm." *Communications of the ACM*, 1970.

[EGL19] Mark Edmonds, Feng Gao, Hangxin Liu, Xu Xie, Siyuan Qi, Brandon Rothrock, Yixin Zhu, Ying Nian Wu, Hongjing Lu, and Song-Chun Zhu. "A tale of two explanations: Enhancing human trust by explaining robot behavior." *Science Robotics*, **4**(37), 2019.

[EGX17] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. "Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles." In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[EHN96] Kutluhan Erol, James Hendler, and Dana S Nau. "Complexity results for HTN planning." *Annals of Mathematics and Artificial Intelligence*, **18**(1):69–93, 1996.

[EKS18] Mark Edmonds, James Kubricht, Colin Summers, Yixin Zhu, Brandon Rothrock, Song-Chun Zhu, and Hongjing Lu. "Human Causal Transfer: Challenges for Deep Reinforcement Learning." In *Annual Meeting of the Cognitive Science Society (CogSci)*, 2018.

[EMQ20] Mark Edmonds, Xiaojian Ma, Siyuan Qi, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. "Theory-based Causal Transfer: Integrating Instance-level Induction and Abstract-level Structure Learning." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[EQZ19] Mark Edmonds, Siyuan Qi, Yixin Zhu, James Kubricht, Song-Chun Zhu, and Hongjing Lu. "Decomposing Human Causal Learning: Bottom-up Associative Learning and Top-down Schema Reasoning." In *41st Annual Meeting of the Cognitive Science Society*, 2019.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

[FGT18] Tesca Fitzgerald, Ashok Goel, and Andrea Thomaz. "Human-Guided Object Mapping for Task Transfer." *ACM Transactions on Human-Robot Interaction (THRI)*, **7**(2):17, 2018.

[FHL16] Lu Feng, Laura Humphrey, Insup Lee, and Ufuk Topcu. "Human-interpretable diagnostic information for robotic planning systems." In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[FN71] Richard E Fikes and Nils J Nilsson. "STRIPS: A new approach to the application of theorem proving to problem solving." *Artificial intelligence*, **2**(3-4):189–208, 1971.

[GA19] David Gunning and David Aha. "DARPA's explainable artificial intelligence (XAI) program." *AI Magazine*, **40**(2):44–58, 2019.

[GGZ20] Xiaofeng Gao, Ran Gong, Yizhou Zhao, Shu Wang, Tianmin Shu, and Song-Chun Zhu. "Joint Mind Modeling for Explanation Generation in Complex Human-Robot Collaborative Tasks." In *Proceedings of International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2020.

[GH80] Mary L Gick and Keith J Holyoak. "Analogical problem solving." *Cognitive psychology*, **12**(3):306–355, 1980.

[GMK99] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: Minds, brains, and how children learn.* William Morrow & Co, 1999.

[GT05] Thomas L Griffiths and Joshua B Tenenbaum. "Structure and strength in causal induction." *Cognitive Psychology*, **51**(4):334–384, 2005.

[GT09] Thomas L Griffiths and Joshua B Tenenbaum. "Theory-based causal induction." *Psychological Review*, **116**(4):661–716, 2009.

[Gun17] David Gunning. "Explainable artificial intelligence (XAI)." *Defense Advanced Research Projects Agency (DARPA)*, 2017.

[HAD18] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. "Explainable neural computation via stack neural module networks." In *European Conference on Computer Vision (ECCV)*, 2018.

[HC11] Keith Holyoak and Patricia W. Cheng. "Causal learning and inference as a rational process: The new synthesis." *Annual Review of Psychology*, **62**:135–163, 2011.

[Hei58] Fritz Heider. *The Psychology of Interpersonal Relations.* Psychology Press, 1958.

[HF11] Thomas Hinrichs and Kenneth D Forbus. "Transfer learning through analogy in games." *AI Magazine*, **32**(1):70–70, 2011.

[HOT06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." *Neural Computation*, **18**(7):1527–1554, 2006.

[HRT13] Marta Halina, Federico Rossano, and Michael Tomasello. "The ontogenetic ritualization of bonobo gestures." *Animal Cognition*, **16**(4):653–666, 2013.

[HS17] Bradley Hayes and Julie A Shah. "Improving robot controller transparency through autonomous policy explanation." In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2017.

[JBD00] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. "Foundations for an empirically determined scale of trust in automated systems." *International Journal of Cognitive Ergonomics*, **4**(1):53–71, 2000.

[Kel99] Deborah Kelemen. "The scope of teleological thinking in preschool children." *Cognition*, **70**(3):241–272, 1999.

[KLH17] James R Kubricht, Hongjing Lu, and Keith J Holyoak. "Individual Differences in Spontaneous Analogical Transfer." *Memory and Cognition*, **45**(4):576–588, 2017.

[KRD18] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. "Textual explanations for self-driving vehicles." In *European Conference on Computer Vision (ECCV)*, 2018.

[KSM17] Ken Kansky, Tom Silver, David A Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, Scott Phoenix, and Dileep George. "Schema networks: Zero-shot transfer with a generative causal model of intuitive physics." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1809–1818. JMLR. org, 2017.

[LDL18] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James Rehg, and Le Song. "Towards black-box iterative machine teaching." In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

[LFD16] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. "End-to-end training of deep visuomotor policies." *Journal of Machine Learning Research*, **17**(39):1–40, 2016.

[LH07] Shane Legg and Marcus Hutter. "Universal intelligence: A definition of machine intelligence." *Minds and Machines*, **17**(4):391–444, 2007.

[LHH20] Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. "A Competence-aware Curriculum for Visual Concepts Learning via Question Answering." *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[LHP15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971*, 2015.

[LLS15] Ian Lenz, Honglak Lee, and Ashutosh Saxena. "Deep learning for detecting robotic grasps." *The International Journal of Robotics Research*, **34**(4-5):705–724, 2015.

[Loc80] A. Lock. *The guided reinvention of language*. Academic Pr, 1980.

[Lom06] Tania Lombrozo. "The structure and function of explanations." *Trends in Cognitive Sciences*, **10**(10):464–470, 2006.

[Lom13] Tania Lombrozo. "Explanation and Abductive Inference." In *The Oxford Handbook of Thinking and Reasoning*. Oxford University Press, 2013.

[LPK18] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." *The International Journal of Robotics Research (IJRR)*, **37**(4-5):421–436, 2018.

[LS04] John D Lee and Katrina A See. "Trust in automation: Designing for appropriate reliance." *Human factors*, **46**(1):50–80, 2004.

[LW85] J David Lewis and Andrew Weigert. "Trust as a social reality." *Social forces*, **63**(4):967–985, 1985.

[LXM17] Hangxin Liu, Xu Xie, Matt Millar, Mark Edmonds, Feng Gao, Yixin Zhu, Veronica J Santos, Brandon Rothrock, and Song-Chun Zhu. "A Glove-based System for Studying Hand-Object Manipulation via Joint Pose and Force Sensing." In *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[LYL08] Hongjing Lu, Alan L Yuille, Mimi Liljeholm, Patricia W Cheng, and Keith J Holyoak. "Bayesian generic priors for causal learning." *Psychological Review*, **115**(4):955–984, 2008.

[LZS18] Hangxin Liu, Yaofang Zhang, Wenwen Si, Xu Xie, Yixin Zhu, and Song-Chun Zhu. "Interactive robot knowledge patching using augmented reality." In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2018.

[Mac42] Colin Maclaurin. *A Treatise of Fluxions: In Two Books. 1*, volume 1. Ruddimans, 1742.

[Mar18] Gary Marcus. "Deep learning: A critical appraisal." *arXiv preprint arXiv:1801.00631*, 2018.

[MBM16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning." In *International Conference on Machine Learning (ICML)*, 2016.

[MG00] Maria Madsen and Shirley Gregor. "Measuring human-computer trust." In *Australasian Conference on Information Systems*, 2000.

[MGK18] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision." In *International Conference on Learning Representations (ICLR)*, 2018.

[MKS15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning." *Nature*, **518**(7540):529–533, 2015.

[MN12] Paula Marentette and Elena Nicoladis. "Does ontogenetic ritualization explain early communicative gestures in human infants." *Developments in primate gesture research*, **6**:33, 2012.

[MP17] Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[Mui94] Bonnie M Muir. "Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems." *Ergonomics*, **37**(11):1905–1922, 1994.

[NC36] Isaac Newton and John Colson. *The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines*. Henry Woodfall; and sold by John Nourse, 1736.

[Nes03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[Pea09] Judea Pearl. *Causality*. Cambridge University Press, 2009.

[PMS16] Emilio Parisotto, Abdel-rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, and Pushmeet Kohli. "Neuro-symbolic program synthesis." In *International Conference on Learning Representations (ICLR)*, 2016.

[QJZ18] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. "Generalized Earley Parser: Bridging Symbolic Grammars and Sequence Data for Future Prediction." In *International Conference on Machine Learning (ICML)*, 2018.

[QLZ20] Shuwen Qiu, Hangxin Liu, Zeyu Zhang, Yixin Zhu, and Song-Chun Zhu. "Human-Robot Interaction in a Shared Augmented Reality Workspace." In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[RW72] R. A. Rescorla and A. R. Wagner. "A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement." *Classical conditioning II: Current research and theory*, **2**:64–99, 1972.

[SB90] Richard S Sutton and Andrew G Barto. "Time-derivative models of Pavlovian reinforcement." In *Learning and Computational Neuroscience: Foundations of Adaptive Networks*. Citeseer, 1990.

[SB98] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[SD88] D. R. Shanks and A. Dickinson. "Associative accounts of causality judgment." *Psychology of learning and motivation*, **21**:229–261, 1988.

[SF15] Aimee E. Stahl and Lisa Feigenson. "Observing the unexpected enhances infants' learning and exploration." *Science*, **348**(6230):91–94, 2015.

[SGG08] Andreas J Schmid, Nicolas Gorges, Dirk Goger, and Heinz Worn. "Opening a door with a humanoid robot using multi-sensory tactile feedback." In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2008.

[Sha91] David R Shanks. "Categorization by a connectionist network." *Journal of Experimental Psychology*, **17**(3):433–443, 1991.

[SHM16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search." *Nature*, **529**(7587):484–489, 2016.

[Sim07] Jeffry A Simpson. "Psychological foundations of trust." *Current directions in psychological science*, **16**(5):264–268, 2007.

[SLA15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust region policy optimization." In *International Conference on Machine Learning (ICML)*, 2015.

[SQA16] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. "Prioritized Experience Replay." In *International Conference on Learning Representations (ICLR)*, 2016.

[Sto95] Andreas Stolcke. "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities." *Computational linguistics*, 1995.

[SV10] David Silver and Joel Veness. "Monte-Carlo planning in large POMDPs." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

[SWD17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347*, 2017.

[SZZ20] Stephanie Stacy, Qingyi Zhao, Minglu Zhao, Max Kleiman-Weiner, and Tao Gao. "Intuitive signaling through an "Imagined We"." In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2020.

[TC97] Michael Tomasello and Josep Call. *Primate cognition*. Oxford University Press, 1997.

[TGK06] Joshua B Tenenbaum, Thomas L Griffiths, and Charles Kemp. "Theory-based Bayesian models of inductive learning and reasoning." *Trends in Cognitive Sciences*, **10**(7):309–318, 2006.

[Thi98] Michael Thielscher. *Introduction to the fluent calculus*. Citeseer, 1998.

[Tom96] Michael Tomasello. "Do apes ape." *Social learning in animals: The roots of culture*, pp. 319–346, 1996.

[Tom10] Michael Tomasello. *Origins of human communication*. MIT press, 2010.

[TPZ13] Kewei Tu, Maria Pavlovskaia, and Song-Chun Zhu. "Unsupervised structure learning of stochastic and-or grammars." In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[TSZ20] Ning Tang, Stephanie Stacy, Minglu Zhao, Gabriel Marquez, and Tao Gao. "Bootstrapping an imagined We for cooperation." In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2020.

[TZ02] Michael Tomasello and Klaus Zuberbühler. *Primate vocal and gestural communication.* MIT Press, 2002.

[WH92] Michael R. Waldmann and Keith J. Holyoak. "Predictive and diagnostic learning within causal models: Asymmetries in cue competition." *Journal of Experimental Psychology: General*, **121**(2):222–236, 1992.

[XLE18] Xu Xie, Hangxin Liu, Mark Edmonds, Feng Gao, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. "Unsupervised learning of hierarchical models for hand-object interactions." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4097–4102. IEEE, 2018.

[YKY18] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. "Representer Point Selection for Explaining Deep Neural Networks." In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[YLF20] Tao Yuan, Hangxin Liu, Lifeng Fan, Zilong Zheng, Tao Gao, Yixin Zhu, and Song-Chun Zhu. "Joint inference of states, robot knowledge, and human (false-) beliefs." In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2020.

[YWG18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[ZGJ19] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. "RAVEN: A Dataset for Relational and Analogical Visual rEasoNing." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[ZJG19] Chi Zhang, Baoxiong Jia, Feng Gao, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. "Learning Perceptual Inference by Contrasting." In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[ZM07] Song-Chun Zhu and David Mumford. "A stochastic grammar of images." *Foundations and Trends® in Computer Graphics and Vision*, **2**(4):259–362, 2007.

[ZNZ18] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. "Interpretable convolutional neural networks." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[ZRH20] Quanshi Zhang, Jie Ren, Ge Huang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. "Mining Interpretable AOG Representations from Convolutional Networks via Active Question Answering." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[ZVM18] Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. "A Study on Overfitting in Deep Reinforcement Learning." *arXiv preprint arXiv:1804.06893*, 2018.

[ZWW20] Quanshi Zhang, Xin Wang, Ying Nian Wu, Huilin Zhou, and Song-Chun Zhu. "Interpretable CNNs for Object Classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[ZZ18] Quanshi Zhang and Song-Chun Zhu. "Visual interpretability for deep learning: a survey." *Frontiers of Information Technology & Electronic Engineering*, **19**(1):27–39, 2018.

[ZZZ20] Zhenliang Zhang, Yixin Zhu, and Song-Chun Zhu. "Graph-based Hierarchical Knowledge Representation for Robot Task Transfer from Virtual to Physical World." In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2020.