Deep Learning, Transparency and Trust in Human Robot Teamwork

Michael Lewis, Huao Li

University of Pittsburgh, School of Computing and Information

Katia Sycara

Carnegie Mellon University, Robotics Institute

Abstract

For Autonomous AI systems to be accepted and trusted, the users should be able to understand the reasoning process of the system (i.e., the system should be transparent). Robotics presents unique programming difficulties in that systems need to map from complicated sensor inputs such as camera feeds and laser scans to outputs such as joint angles and velocities. Advances in Deep Neural Networks are now making it possible to replace laborious handcrafted features and control code by learning control policies directly from high dimensional sensor inputs. Because Atari games, where these capabilities were first demonstrated, replicate the robotics problem they are ideal for investigating how humans might come to understand and interact with agents who have not been explicitly programmed. We present computational and human results for making DRLN more transparent using object saliency visualizations of internal states and test the effectiveness of expressing saliency through teleological verbal explanations.

1. Introduction

Teamwork is a set of interrelated reasoning, actions and behaviors of each team member that adaptively combine to fulfill shared team goals (Morgan et al., 1986). Experimental evidence from high performance human teams resulted in a set of drivers of team effectiveness (Salas et al., 2005, 2008). These drivers are: *team leadership*, *mutual performance monitoring*, *backup behaviors* (ability to anticipate other team members' needs and shift work and cognitive workload to achieve appropriate balance), *adaptability* (ability to adjust strategies and actions based on dynamic changes in mission and environment), *team orientation* (taking other's behavior into account during group interactions and belief on team goals over individual goals), *shared mental models* (organizing knowledge structure of the relationship between tasks and how the team will perform them), *closed loop communication* (reliable exchange of information) and finally and most crucially mutual *trust* (the shared belief that team members will perform their roles and protect the interests of their teammates). As technology enables

increased machine autonomy, human-machine teaming could acquire the same characteristics as human-human teaming.

Besides being an important ingredient of teamwork, trust has been found to be important in human use of automation: people tend to rely on automation they trust and not use automation they do not trust. This has generated sustained interest in conceptualizations of trust and its relation to human interaction with automation. Trust has been defined by Mayer et al. (1995) as "*The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that party.*" In the context of trust in automation the above definition has typically been interpreted as a human's willingness to rely on automation to perform some task. Research in human interaction with automation, which we believe is also valid for human interaction with autonomy, has found that the human may fail to use the automation when it would be advantageous, called *disuse or under-reliance*, fail to monitor it properly when in use, or accept its recommendations and actions when inappropriate, called *misuse or over-reliance* (Lyons & Stokes, 2012). Lee & Moray (1992, 1994); Muir & Moray (2013); Lewandowsky et al. (2000) have shown that trust towards automation can mediate reliance. Additionally, operator trust has been found to *vary dynamically*, accumulating over periods of successful performance, then declining sharply when failures or poor performance are encountered (Lee & Moray, 1994; Gao & Lee, 2006; Xu & Dudek, 2015).

In social robots, incorrect trust calibration can lead to extreme overtrust as demonstrated in Robinette et al. (2016) where participants followed the directions of a demonstrably dysfunctional robot to evacuate a smoke filled room with detrimental results, including getting directed into closets rather than following exit signs. In other studies reviewed by Schaefer (2013) trust in social robots increased with matches between robot appearance and user expectations and were generally higher for less socially competent robots.

Another characteristic that contributes to both teamwork effectiveness and trust is *transparency*. Systems that are more transparent in conveying their reasoning should be more trusted, since they would be more easily understood by their users (Simpson et al., 1995; Sycara et al., 1998; Lewis, 1998; Lewis & Heidorn, 1991) and hopefully their users would be able to better judge their capabilities, thus improving their trust calibration. Additionally transparency in a human-robot context can be viewed as a method to establish shared intent and shared awareness between a human and a machine (Lyons, 2013). Although there are multiple definitions of agent transparency (Chen et al., 2014; Lyons & Havig, 2014), we use, with minor variation, the definition proposed by Chen et al. (2014): "Agent transparency is the quality of an interface (e.g., visual, linguistic) pertaining to its abilities to afford an operator's comprehension about an intelligent agent's intent, performance, future plans, and reasoning process". The goal of transparency is not to relay all of a system's capabilities, behaviors, and decision-making rationale to the human. Ideally, agents should relay clear and efficient information as succinctly as possible, thus enabling the human to maintain a proper understanding of the system in its tasking environment. The human factors and computational literature (Kim, 2015; Mercado et al., 2016; Ribeiro et al., 2016) has pointed out the need for system transparency as a way to increase trust in the system.

As agents become more sophisticated and independent via learning and interaction, it is critical for their human counterparts to understand their behaviors, the reasoning process behind those behaviors, and the expected outcomes to properly calibrate their trust in the systems and make appropriate decisions (de Visser et al., 2014; Lee & See, 2004; Mercado et al., 2016). When interacting with autonomous intelligent agents, people tend to regard them as intentional individuals and explain their behaviors in terms of interpersonal relationships (de Graaf & Malle, 2017). That requires an explainable agent to clarify its actions by offering reasons of beliefs, desires, and intentions (Langley et al., 2017). Indeed, past studies have shown that humans sometimes question the accuracy and effectiveness of agents' actions due to the human's difficulties understanding the state/status of the agent (Bitan & Meyer, 2007; Seppelt & Lee, 2007; Stanton et al., 2007) and the rationales behind the behaviors (Linegang et al., 2006).

In recent years automation and autonomous system have increasingly relied on Machine Learning. ML systems are typically used for two broad types of problems. First, they are used for classification, which relies on supervisory methods with ground truth given by labelled data, and produces a judgment as to whether an input belongs to a particular class. Such systems have become ubiquitous in almost all areas of human endeavor, such as Web services, health care, education, insurance, law enforcement and defense (Lipron, 2013). Machine learning algorithms make important decisions in our interactions influencing the news we see, our finances (who gets a loan, or a particular line of credit), careers (algorithms often filter job applications). Courts have employed algorithms to predict the probability that an individual relapses into criminal behavior (Choudlechova, 2016). Neural Networks used in classification have revolutionized computer vision and natural language understanding. Second, ML systems solve sequential decision making problems that use mainly unsupervised methods to produce a series of decisions that would give an optimal reward. These systems work using Deep Neural Networks for Reinforcement Learning (DRL) where an agent explores the space of possible strategies in an environment and receives feedback (positive or negative) on the outcome of choices it makes. Given a particular domain, exploration allows the agent to form a strategy, called policy, that allows it to generate and follow a sequence of actions to maximize its payoff (see section 4.1 for related work in this area). The last few years have witnessed a rapid growth of research and interest in the domain of deep Reinforcement Learning (RL) due to the significant progress in solving RL problems (Arulkumaran et al., 2017). Deep RL has been applied to a wide variety of disciplines ranging from game playing, to robotics to dialogue systems (Silver et al., 2017; Mnih et al., 2015b; Levine et al., 2016; Kraska et al., 2018; Williams et al., 2017; Choi et al., 2017).

Self-driving cars that are poised to be deployed in the near future increasingly employ AI-based object recognition software, and military autonomous systems may be called to make decisions that cost civilian lives. These systems are more opaque than their predecessors since they rely almost exclusively on learning form data in order to shape their conclusions and behavior, as opposed to model-based systems that are more understandable and often have algorithms that provide formal guarantees of performance. The inner logic and reasoning of DRL systems are opaque and difficult to be understood even by their own designers.

The opacity of these systems is becoming increasingly problematic and therefore there is increasing clamor for transparency in the machine learning community as well (Kim, 2015; Ribeiro et al., 2016). This need for system transparency has resonated with policy makers as well. For example, new regulations in the European Union propose that individuals affected by algorithmic decisions have a *right to explanation* (Goodman & Flaxman, 2016).

In this chapter, we discuss the intertwined notions of trust and transparency in the context of human-agent teamwork. Here agents can be autonomous systems that engage in sequential decision making processes and control, can be computational processes that comprise robotic components, such as a vision system for the robot, or provide information and suggestions to humans. We will present a review of trust and transparency in the human factors literature emphasizing the need to consider transparency in all dimensions of the trust construct, and then we will turn our attention to the challenges of transparency for Deep Learning algorithms. We will then present our own work on transparency for Deep Reinforcement Learning, followed by conclusions and open research problems.

2. Factors Affecting Trust in Automation

It is generally agreed in the psychological literature that trust is best conceptualized as a *multidimensional psychological attitude* involving beliefs and expectations about the trustee's trustworthiness derived from experience and interactions with the trustee (Jones & George, 1998). Although in the literature, the number and concepts in the trust dimensions vary Jian et al. (2000); Madsen & Gregor (2000), Lee & See (2004) point out a broad consensus on 3 dimensions they label Purpose (what the automation is supposed to do), Process (how automation goes about fulfilling its purpose) and Performance (actual performance). Moreover, it has been found that trust is dynamic (Lee & Moray, 1994; Lee & See, 2004; Nam et al., 2017, 2019) and a human's history of interaction with automation affects future behavior indirectly through changes in trust.

The most important factors that affect trust in automation are:

(a) *System reliability:* Prior literature has provided empirical evidence that as automation reliability declines, human trust declines and vice versa (Hancock et al., 2011; Moray et al., 2000; Moray & Inagaki, 1999; Riley, 1994; Parasuraman & Manzey, 2010). Additional studies have shown that imperfect (unreliable) automation can have adverse effects on reliance and compliance (Dixon & Parasuraman, 2006; Meyer, 2004), and overall system performance de Visser et al. (2006); Dzindolet et al. (2003). Moray et al. (2000) showed that declining system reliability can lead to systematic decline in trust and trust expectations, and most crucially, these changes can be measured over time.

(b) *System predictability:* Research has shown that when people have prior knowledge of faults, these faults do not necessarily diminish trust in the system (Riley, 1994; Lewandowsky et al., 2000), possibly due to the fact that knowing how the automation may fail reduces the uncertainty and consequent risk associated with use of the automation.

(c) *System intelligibility and transparency:* Prior research on trust in automation found that providing human operators with information related to the reliability of an automated tool promoted more optimal reliance strategies on the tool (Lyons & Havig, 2014). Further, information related to the limitations of an automated tool aids in trust recovery following errors of the automation (Choudlechova, 2016). Given that reliability of performance is the biggest determiner of trust (Hancock et al., 2011),providing additional information about performance such as knowledge of results Beck et al. (2007); Dzindolet et al. (2002) or confidence judgements Dadashi et al. (2013) have also been widely used to increase trust through greater transparency. The Human Factors literature has paid less attention to the Purpose and Process components of trust, most frequently assuming Purpose to be evident. Attention to Process has concentrated mostly on developing displays for mission-based systems to show the state of the system in more transparent ways to allow the human to understand the system (Wang et al., 2016) or to improve trust calibration (Lyons et al., 2016).

(d) *Level of Automation:* Another factor that may affect trust in the system is its level of automation (i.e., the degree to which the system acts on its own). Since higher levels of automation are more complex, *thus potentially more opaque* to the operator, higher levels of automation are frequently (Calhoun et al., 2009; Amato et al., 2011; Nam et al., 2019; Kira & Potter, 2009) found to engender less trust. To overcome this deficit transparency becomes more important when the system is more autonomous. (Oh et al., 2015) In summary, prior literature indicates that a high level of autonomy may cause the operator to undertrust the system. These implications are echoed in the distrust and clamor for transparency for systems based on Deep Neural Networks.

3. Trust and Transparency in Human-Autonomy Teaming

Since transparency is an important ingredient of trust, and since trust is multi-dimensional, we believe that work on transparency should span all trust dimensions, in particular purpose, process and performance. Additionally, since factors such as Degree of Autonomy modulates trust, its influence should also be studied. Transparency can be viewed as the degree to which automation conveys the basis of its behavior. Parasuraman et al. (2000) proposed a model characterizing automation as a sequence of stages which were reduced to two by Wickens (2018): 1) situation assessment and 2) action choice and execution. Transparency effects can usefully be organized by these stages of autonomy Wickens (2018) and dimensions of trust Chen et al. (2014).

Figure 1 presents constituents of transparency that we briefly discuss in this section in the context of the trust dimensions of performance, purpose and process (also depicted in the figure). Additionally, the figure sketches our explanation model for DRL agents and shows its relationship with transparency and trust.

Transparency of performance (execution stage) such as system reliability is often assumed to be directly observable to users yet may be perceived inaccurately. Biros et al. (2004), for example, found trust and reliance to be influenced by cover stories. Other studies (Beck et al., 2007) have found supplying knowledge of results to lead to better trust calibration and improved reliance. Annotating decisions with confidence judgments is another widely used technique for providing greater insight into system performance. This has been done in a variety of ways (e.g., by providing probabilities

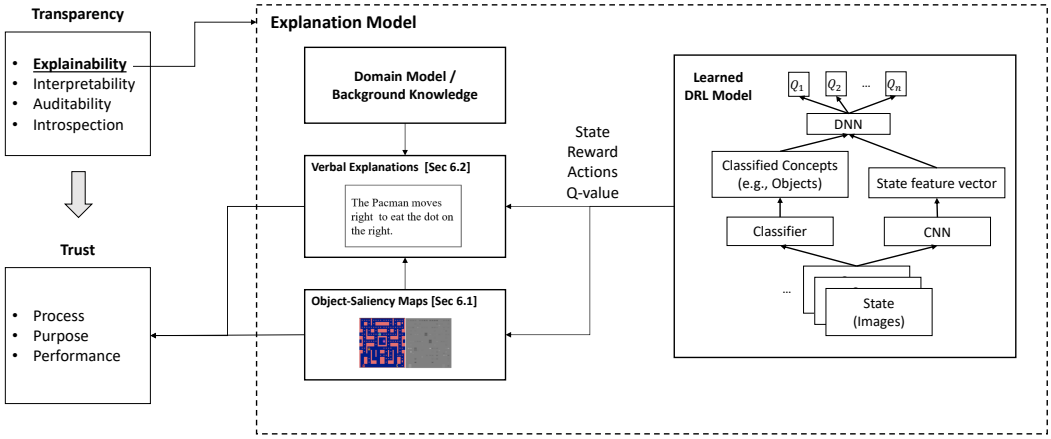


Figure 1: A block diagram of the visual and verbal explanations offered by our model of DRL agents and its relationship with transparency and trust. The model is reported in Section 5.

of success) (Lyons et al., 2016; Wang et al., 2016; Dadashi et al., 2013), confidence in detection (Dadashi et al., 2013) or expected reward from selected robot action (Wang et al., 2016).

Transparency of purpose: For pre-programmed automation the system's purpose is static and known a priori (purpose transparency was baked in) to the user. As systems become increasingly autonomous and thus learning and adaptive, their *intent/purpose* may change over time and may depend on environmental and social factors. Therefore, the system's intent must be communicated to the human to help the human understand and anticipate system activities. To do this, the system must be able to **introspect**, (i.e., be aware of its own goals, preferences and what factors bring change to them). For example, a system might change its intent in response to a perceived change in intent of the human with whom the system is teaming.

In experiments with humans, the system's purpose is almost always informally conveyed through instructions or the demand characteristics of an experiment, however, when explicitly manipulated, as in Sadler et al. (2016) effects on trust and reliance were found. More commonly, experiments involving transparency of automation do not discriminate between stages of automation or form of explanation. They instead mix categories often in an additive manner such as Verberne et al. (2012) which started with an unsupported recommendation in the control condition (execution stage), increased transparency of performance by conveying expected risk in the second condition, and added an explanation (situation assessment stage) involving process in the third.

Transparency of process (situation assessment stage): While trust and the decision to rely on or comply with an autonomous agent may depend on knowledge of its purpose or reliability, the reciprocal interactions needed for teamwork depend crucially on the ability to predict and coordinate behavior (i.e., knowledge of process). In the current human factors literature, transparency of process involves conveying infor-

mation to the human about the system's decision making process to support operator situation awareness (Chen et al., 2014; Lee & See, 2004). Increased information, such as explanations by an agent, may have a complicated effect on operators' workload. Although a human may need to process more information from an agent with high transparency, this information might be desired in order to achieve better performance. Therefore, additional information may help operators to understand the intentions and behaviors of an agent and correct their expectation and trust in it without increasing workload (Chen & Barnes, 2014). Recent work has confirmed the above argument in multi-unmanned-vehicle control task scenarios (Mercado et al., 2016).

The reasoning (transparency of process) of pre-programmed automation, though not necessarily known to the user a priori, was easier to convey. For example, a rule-based system can display its set of rules to allow a human to see its reasoning. A very early example of this was Teiresias (Davis, 1978) that explained its reasoning by showing the rule chain that led to its conclusions. Even for more sophisticated systems that may use (e.g., Bayesian processes, or decision trees), the system's reasoning is relatively easy to convey.

The broad use of purely data-driven AI/ML techniques raises additional challenges that necessitate a more refined look at the concept of transparency and trust. If the data is biased, the system may come to the wrong conclusions although its algorithm operates correctly. In a well known example of such behavior, the data used by an image classification system inadvertently included images of wolves only in snowy backgrounds (Ribeiro et al., 2016). When the system was given an image of a wolf in a non-snowy background, it misclassified it as a dog. Only after the system designers searched for the cause, they discovered it was due not to a faulty algorithm but to biased data. In this particular case, recognizing the mistake was easy (finding the cause was not so easy), however in other situation (e.g., system decisions on recidivism or extending credit), even recognition of faulty system conclusions is not easy. Besides data biases, adversarial data manipulation is also a major source of concern. Convolutional Neural Networks (CNNs), used for image classification, are susceptible to adversarial manipulation that causes them to misclassify images that seem only imperceptibly perturbed to a human (Szegedy et al., 2013). Their paper shows examples of images that were slightly perturbed by adversarial noise (imperceptible to the human eye) including a yellow school bus and a white dog that were both misclassified as ostriches!. Thus, for these systems to be trusted, they should be transparent in terms of being **auditable** as to possible biases in the data and robust to adversarial manipulation. This notion of auditability as a characteristic of transparency has also been found in the human factors literature where increases in compliance was found from providing means of *verifying* automation decisions (Bliss et al., 1996). Making progress towards discovering and correcting biases and also discovering and protecting systems from adversarial data manipulation are open research problems in the AI/ML community.

Besides auditability of decisions, transparency can also be considered in terms of **interpetability** and **explainability**. Interpretability enables understanding of the mechanics of the algorithm, whereas explainability enables producing post hoc explanations or predictions of the model without necessarily elucidating the mechanisms by which the model works (Lipton, 2016). Both interpetabilty and explainability, like trust and transparency, also lack formal definitions and characterizations. There is in-

creasing interest in the AI community of providing those and developing techniques for understanding them. In section 5, we present our own work on transparent/explainable Deep Reinforcement Learning using object-based saliencey maps.

3.1. Types of Explanations

Explanations are conventionally categorized as:

- Teleological or functional (WHY): Explanations for transparency may be teleological (why) which may be incomplete, conveying simply relevant features of the situation contributing to an automation decision (Wright et al., 2016) or complete in supplying both features and logic behind the decision. Teleological explanations are preferred by humans (Lombrozo, 2006) who are strongly predisposed to attribute causality (Thines, 1991) even in its absence. We have strong predispositions to attribute causality as shown in Michotte's classic experiments reported by Thines (1991) in which subjects interpreted the motions of abstract geometric objects as elastic collisions or animate behaviors such as chasing or leading depending on timing and proximity of objects. Explanations of more abstract behavior typically appeal to causes as well, with knowledge of general patterns constraining which causes are judged probable and relevant Lombrozo (2006) in much the way timing and proximity do for perception. When explanations are judged for quality the presence of a general pattern is typically preferred to probability judgments alone while this pattern is required to be both general and relevant (appropriate to the actors and event). So, for example, an explanation that "a robot turned left to avoid a collision" would be preferable to an explanation that the turn was taken to increase the robot's long term probability of reaching a goal, although both might be correct. Similarly, citing the presence of a wall (often present preceding turns) or distance from door (a serendipitous feature of the robot's history) as cause would fail the probable and relevant test since walls were often present when turns did not occur (not relevant) and distance from door does not reference a humanly known general pattern for evoking turns (not probable). Moreover, this explanation implicitly includes the the domain knowledge that collisions are undesirable, that the robot had learned to avoid collisions, and that collision is threatened by a trajectory leading the robot into the wall.

- Efficient or mechanistic (HOW): the primary source of an action "the robot's programming caused it to turn". Although this explanation has implicit the state of the world at the time of the turn, it is not relevant or useful to the human. Mechanistic explanations can drill down by chaining together a series of intermediate steps so the answer might be elaborated to include that what the robot sensed caused its programming to produce an observed preference value leading it to turn.

- Formal or constitutive (WHAT): expressing a necessary aspect of an object or event or conveying part-whole relationships. For example, the What explanation of the robot taking a turn may be, "The robot took 5 steps forward, turned its orientation to the left, took 5 steps forward".

From this perspective teamwork involving robots trained through DRL presents a dilemma. An understanding of process needed to predict behavior, remains hidden within the layers of the network. The accuracy of our predictions of robot actions therefore, depend on the correspondence between our own attributions of causality and the policies the robot has learned. As summarized by Lombrozo (2006) these attributions are predominantly based on probability and relevance. In the case of the game Ms. Pac-Man, for example, relevance would revolve around the rules of the game in which Pac-Man gains points by eating pellets and avoiding ghosts. The DRL player, however, does not benefit from predefined relevance and learns from scratch to find the optimal policy through trial and error. Therefore, while the resulting policy has been trained to win the game it will not necessarily choose the same actions as a human, making attribution difficult. Recently, the AI community has studied the effects of including domain knowledge in the form of relevant features. Although this may limit generality of learning, since some features may be predetermined, it may help system explainability.

In the context of teamwork, not only should the system be transparent to the user, but also *the human should be transparent to the system*. In other words, the system should be able to make inferences about human intent (purpose), human beliefs and how these beliefs may lead to actions (process) prediction of human actions (performance). Such understanding on the part of the system is equivalent to the system formulating a *Theory of Mind* of the human that would allow it to adapt to human's behavior thus improving teamwork performance. Such a Theory of Mind will allow the agent to better understand why a human may be taking a particular action, understand when the human may have false beliefs (e.g., due to lack of information to environment changes) and inform the human of missing information that the agent may have, so as to correct the human's false beliefs.

4. Background on RL and Deep RL

Reinforcement learning solves the sequential decision making problems by learning from experience. In Reinforcement Learning (RL), an agent interacts with an environment ε over discrete time steps and receives feedback (rewards) on the outcome of choices is makes. Given a state, the agent selects actions in order to maximize future rewards. In the RL setting the problem can be modelled as a Markov Decision Process (MDP) represented by the 5-tuple (S,A,T,R,γ), where S is the state space, A is the action space, $T(s'|s,a)$ is the state transition probability function, $R(s,a)$ is the reward function and $\gamma \in [0,1]$ is the discount factor. Due to stochasticity, a policy $\pi : S \rightarrow A$ maps every state to a distribution over actions. In the time step t, the agent receives a state $s_t \in S$ and selects an action $a_t \in A$ according to its policy π, where S and A denote the sets of all possible states and actions respectively. After executing the action, the agent receives a scalar reward r_t and enters the next state s_{t+1}.

The goal of the agent is to choose actions to maximize its rewards over time. In other words, the action selection implicitly considers the future rewards. The total discounted return is defined as $R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau$ where $\gamma \in [0,1]$ is a discount factor that trades-off the importance of recent and future rewards.

Policy based methods directly model the policy (Williams, 1992), while in value-based RL methods, the action value (a.k.a., Q-value) is commonly estimated by a function approximator, such as a deep neural network (Mnih et al., 2015a). The actor-critic Sutton & Barto (1998) architecture is a combination of value-based and policy-based methods.

The value function $V^{\pi}(s_t)$ is the expected discounted sum of rewards by following policy π from state s_t at time t, $V^{\pi}(s_t) = E[\sum_{i=0}^{T} \gamma^i r_{t+i}]$. Similarly, the Q-value (action-value) $Q^{\pi}(s_t, a)$ is the expected return starting from state s_t, taking action a and then following π. The Q-value function can be recursively estimated using the Bellman equation $Q^{\pi}(s_t, a) = E[r_t + \gamma \max_{a'} Q(s_{t+1}, a')]$ and π^* is the optimal policy which achieves the highest $Q^{\pi}(s_t, a)$ over all policies π.

In the reinforcement learning community the action value function Q is computed via a typically linear function approximator, but in Deep Reinforcement learning, the function approximator is a (nonlinear) neural network. Deep neural networks have been recently applied in reinforcement learning (RL) to achieve human-level control policies in various challenging domains. The rich representations given by a deep neural network improve the efficiency of reinforcement learning (RL) at the expense of requiring a vast amount of training data. If a high fidelity simulator is available (e.g., Zhu et al. (2016) which describes the appearance of the real-world as closely as possible), such data can be easily generated.

The advantages of deep RL are (a) the features do not need to be hand-crafted but are learned during training, (b) deep neural networks have shown superior performance in many challenging domains, and (c) the algorithm is model-free (i.e., it solves the reinforcement learning task directly using sample data without explicitly estimating the reward and transition dynamics).

The challenges of this approach are: (a) large amounts of data are needed, (b) RL with nonlinear function approximator as a neural network could be unstable (or even diverge). This is due to different causes, such as the correlation present in the sequence of observations, the fact that small updates to the value function may significantly change the policy, and the existence of correlations between action values (Q) and the target reward values

DRL differs from image classification in a number of ways: In classification, for each image, the system has to predict its label whereas for DRL the system must learn a function that maps state-value pairs to Q-values that capture the dynamics of the system over time. If the state in DRL is a single image, the dynamics of the system are not captured. Therefore, a *state must be represented by a sequence of images*. Current works on DQNs for Atari games use only 4 previous screens as input to learn the reward for each state.

Although Deep RL is much more challenging than classification, in the past few years a new variant of Deep RL has been developed and tested mainly on Atari games (Mnih et al., 2013) where the states are the input screens of the game. The agent chooses an action from the possible control actions (e.g., up/down/left/right/). After that, the agent receives a reward (how much the score increases or decreases) and the next image input. The deep RL agent, only reasoning on the image pixels and the game scores show performance comparable or higher to an expert human player (Mnih et al., 2015b). However, producing explanations has been out of scope for Deep RL until

recently but interest in interpetability and explainability is increasing (Lipton, 2016).

Developing methods to enable autonomous agents to be transparent is very challenging, because ease of transparency seems to be inversely proportional to agent sophistication. However, DNNs are extremely opaque (i.e., they cannot produce human understandable accounts of their reasoning processes or explanations). Therefore, there is a clear need for deep RL agents to dynamically and automatically offer explanations that users can understand and act upon.

4.1. Related Work on Computational Models

The literature on deep reinforcement learning is fast increasing. Here we briefly review work that is most relevant to this chapter. Multiple deep RL algorithms have been developed to incorporate both on-policy RL such as Sarsa (Rummery & Niranjan, 1994), actor-critic methods (Sutton & Barto, 1998) and off-policy RL such as Q-learning (Watkins, 1989), or combination of RL and experience replay memory (Mnih et al., 2015a; Riedmiller, 2005). A parallel RL paradigm (van Hasselt et al., 2015) has also been proposed to reduce the heavy reliance of deep RL algorithms on specialized hardware or distributed architectures. The deep Q-network (DQN) model proposed in Mnih et al. (2015a) combines Q-learning with a flexible deep neural network. More specifically, recent work has found outstanding performance of deep reinforcement learning models on Atari 2600 games using only raw pixels to make game control decisions (Mnih et al., 2015a). DQN can reach human-level performance on many of Atari 2600 games. However, DQN suffers from substantial over-estimation in some games. van Hasselt et al. (2015) thus proposes Double Q-learning algorithm that can be generalized to work with large-scale function approximation. A dueling network architecture (Wang et al., 2015) has been proposed to decouple the state-action values into state values and action values. The experiments of Mnih et al. (2016) show that the actor-critic (A3C) method surpasses the current state-of-the-art in the Atari game domain. In contrast to Q-learning, A3C is a policy-based model that learns a network action policy. However, for game settings with many objects where each object has a different role in reward computation, A3C does not perform very well. Therefore, Lample & Chaplot (2016) propose a method that augments performance of reinforcement learning by exploiting game feature information.

In human-robot interaction and teaming, robots and humans must adapt to one another. This requires the robots to maintain a computational cognitive model of their human co-workers in the task environment. There is a body of work on making intelligent robots able to 1) adapt to physical human behaviors (Liu et al., 2016), 2) infer human's intent (Hadfield-Menell et al., 2016; Dorsa Sadigh et al., 2017) and 3) shape the way how humans reason about robots (Huang et al., 2019; Zhou et al., 2017; Pezzulo et al., 2013). Previous research assumes the human is a perfect collaborator and uses Bayesian inference to predict the human's next goal in order for the robot to adapt accordingly (Liu et al., 2016). Subsequent work by Hadfield-Menell et al. (2016) extends the scope to situations where robots do not know the human operators' reward function and need to learn it over the course of interaction. Inverse Reinforcement Learning (Ng et al., 2000; Ramachandran & Amir, 2007) approaches this problem from a passive learning perspective like learning from demonstrations offline, while in Cooperative Inverse Reinforcement Learning (Hadfield-Menell et al., 2016) the human

helps the robot learn by making his/her behavior more transparent. Alternately, by maintaining a model of human mental states, the robot may take informative actions to communicate its goals (Pezzulo et al., 2013), reward functions (Huang et al., 2019), or confidence levels (Zhou et al., 2017) to humans.

Ideally, we would like to develop techniques for DRL explanations that explain (a) why an action was taken (teleology), (b) why this action was taken as opposed to other ones (counterfactuals), and (c) why a sequence of actions was taken (plans).

5. Object-Saliency Based Explainability in Deep RL

Most current work around interpretability in deep learning is based on local explanations (i.e., explaining network predictions for specific input examples (Lipton, 2016)). Saliency maps are used to generate local explanations. Saliency maps generally use gradient-like information to identify salient parts of the image and highlight important regions of the input that influence the output of the neural network. Zahavy et al. (2016) use the Jacobian of the network to compute saliency maps on a Q-value network. Perturbation based saliency maps using a continuous mask across the image and also using object segmentation based masks have been studied in the context of deep-RL (Greydanus et al., 2017). Our object-based saliency method (Iyer et al., 2018; Li et al., 2017) that we present in 5.1 belongs to this category.

In contrast, global explanations attempt to understand the mapping learned by a neural network regardless of the input. There are additional difficulties with global explanations since there are problems of generalization and memorization. Recent findings suggest that deep RL agents can easily memorize large amounts of training data with drastically varying test performance and are vulnerable to adversarial attacks (Zhang et al., 2018a,b; Huang et al., 2017). We have explored interpetability using global explanations in Annasamy & Sycara (2019). Our method aims to understand aspects of the input space (images) that are captured in the latent space across inputs. In particular, given a particular action (e.g., Pac-Man goes left) and expected returns (e.g., rewards of 50 points), our method tries to understand visually which would be the features of the corresponding states. This could help in implicitly identifying underlying relations between the entities in these states, (e.g., the network might have learned that presence of a ghost in the vicinity of Pac-Man would be a had thing that must be avoided). However, our results suggest that the features extracted by the convolutional layers are extremely shallow and can easily overfit to trajectories seen during training, rather than generalize to trajectories that would be in conformance with real relations among entities.

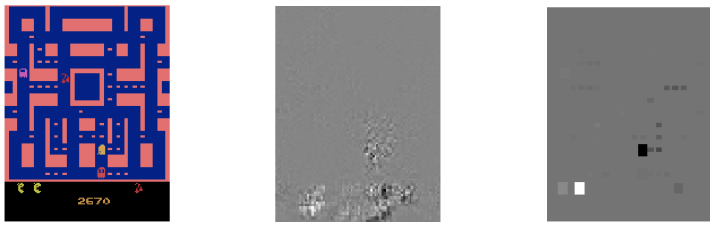
In the following section we present a short description of our recent work on explainability in DRL, using the game of Pac-Man. In this work, we have explored bot visualization and text as means of producing local explanations of agent (Pac-Man) behavior.

5.1. Visual Explanation

In Atari games, the DRL network is supposed to *implicitly* learn all relevant features that capture the agent's reasoning and behavior. However, this does not aid explainability to humans. Since humans recognize objects, our idea was to enhance the

neural network with object recognition ability as a way to get a handle on relevance of different objects for the DRL agent's decision making. We first used template matching, a computer vision technique (Brunelli, 2009) to recognize objects in images. The technique works by taking a template image (the patch) and sliding it through a source image (up to down, left to right) and calculating the current source image similarity to the template image. After object recognition, we used *object channels* to incorporate features of objects in the input images to the DRL network.

The object channels as well as the original image are given to the network as input. The network outputs (predicts) Q-values for each action. This method can be used incorporated into different existing deep reinforcement learning algorithms, such as DQN or ACC.



(a) Screenshot of the State

(b) Pixel Saliency Map

(c) Object Saliency Map

Figure 2: An example of original state, corresponding pixel saliency map and object saliency map produced by a double DQN agent in the game "Ms. Pac-Man."

Our method to provide transparency for Deep Neural Networks is called *object saliency maps*. A saliency map highlights regions of the input that, if changed, would most influence the output (Simonyan et al., 2013). Saliency maps is an ex post facto local explanation method. This means that after the network has been trained, query images can be input that may help the user predict what the network was paying attention to (ie. "explain) so as to help the user understand the system's reasoning and also possibly predict the system's next decision. Object saliency maps provide visualization of the decisions made by RL agents. These visualizations aim to be intelligible to humans. To generate intelligible visualizations that would help with explanations of DQN agent behaviors, we need to determine which pixels the model pays attention to when making a decision (Simonyan et al., 2013). Another interpretation of computing pixel saliency is that the value of the derivative indicates which pixels need to be changed the least to affect the Q-value.

However, pixel-level representations are not intelligible to people. Figure 2(a) shows a screenshot from the game Ms.Pac-Man. Figure 2(b) is the corresponding pixel saliency map produced by an agent trained with the Double DQN(DDQN) model. The agent chooses to go right in this situation. Although we can get some intuition of which area the deep RL agent is looking at to make the decision, it is not clear what objects the agent is looking at and why it chooses to move right. On the other hand, Figure 2 makes more intelligible the objects that the DQN is paying attention to. To understand the influence of objects on agent decisions, we need to rank the objects

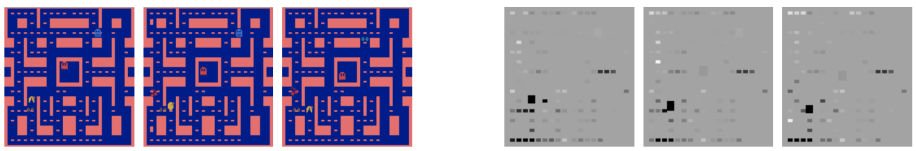
in a state s based on their effect on $Q(s,a)$. In the game of Pac-Man, there are static pellets/dots that Pac-Man eats to get points. There are ghosts that chase and eat the Pac-Man, which finishes the game and gives a large number of negative points. There are super-pellets and cherries that appear dynamically, and if Pac-Man eats them then it gets more points. Moreover, if Pac-Man eats a cherry then, the ghosts become edible for some time, so if the Pac-Man manages to eat an edible ghosts it gets a very high reward (many points).

For each object O found in s, we mask the object with background color to form a new state s_o as if the object does not appear in this new state. We calculate the Q-values for both states, and the difference w of the Q-values actually represents the influence of this object on $Q(s,a)$. So, if w is positive the object has a positive influence which means the *the object gives positive future reward to the agent* (the positive objects are shown in dark in the saliency maps). Negative w represents "bad" object since after we remove the object, the Q-value gets improved.

Figure 2(c) shows an example of the object saliency map that clearly shows which objects the model is paying attention to and the relative importance (via shading) of each object.

5.1.1. Human Experiments

In order to test whether the object saliency map visualization can help humans understand the learned behavior of Pac-Man, we performed an initial set of experiments. The goals of the experiment were to: 1) test whether object saliency maps contain enough information to allow humans to match them with corresponding game scenarios, 2) test whether participants could use object saliency maps to generate reasonable explanations of the behavior of the Pac-Man and 3) test whether object saliency maps allow participants to correctly *predict* the Pac-Man's next action. This requires a deeper causal understanding of what may influence the Pac-Man in his decisions.



(a) Screen-shots (b) Object Saliency Maps

Figure 3: An example of the stimulus materials participants saw on trial 9 of the prediction task. 75% participants in the screen-shot group thought Pac-Man would go left to eat the cherry at the left side. 60% participants in the object saliency maps group predicted the Pac-Man would keep going down for the dark elements (the pellets) below.

Experiments were conducted in a graduate and undergraduate Human Factors class. The forty participants were approximately equally divided by gender and between 20-29 years old. The Matching and Prediction tasks were presented sequentially over a period of 30-45 minutes.

Matching Task In each trial, the participants are shown twice, a 5-second video clip of Pac-Man gameplay generated by O-DDQN. During the video clip, Pac-Man de-

cides and takes particular actions. The last decision made produces the crucial movement of the clip (e.g., Pac-Man moves right), with the clip ending just after the crucial movement. Three frames from the object saliency map are then shown to participants (see Fig. 3(b)). The center frame is the frame where the Pac-Man makes the crucial decision and the other two are frames from before and after that moment. In the task, participants are asked to judge whether the saliency maps accurately represent the video they just saw. In the matching cases, the saliency maps indeed were generated from the video clip the participant saw. In the non-matching cases, the three saliency map frames were generated from a different video clip. In distractor/non-matching clips, the Pac-Man occupies the same area of map as in the target video, but makes different movements. This is done to avoid the case where the participants solely focus on the location of the Pac-Man as a matching criterion, disregarding the movements and environmental factors.

Following the match decision, if the participants' answer is "match", they are asked to give an explanation for the Pac-Man's movements based on the video and saliency maps. In other words, participants are asked to provide a teleological explanation explaining 'why' Pac-Man acted as she did. For example, "Pac-Man moved up to eat more energy pellets while avoiding the ghost coming from below."

The matching task consisted of 2 training trials and 20 test trials, half (10 trials) presenting matched video and saliency maps, the other half presenting non-matched pairs in a single randomly ordered sequence. Dependent variables were correctness of matches and agreement between explanations and saliency maps.

Prediction Task In each trial, the participants are shown a video clip not used in the matching task. Each clip ends at the point where the Pac-Man must choose a crucial move. The participants are divided equally into two experimental conditions. In the screen-shot condition, after the video clip, participants see 3 actual screen-shots from the video ending before the crucial move is taken. In the object saliency map condition, the participants see three object saliency map frames (corresponding to the screen-shot frames) after viewing the video clip (see Fig. 3). At the decision point in the third frame Pac-Man's choices (up, down, left, right) may be limited by barriers indicated on the response forms. Participants are asked to predict Pac-Man's movement among the feasible directions based on the three previous frames (screenshots or saliency maps), and then give an explanation for their prediction which includes their judgment as to which elements of the game influenced the Pac-Man's decision (indicating these elements by circling them on a hardcopy of the screenshot or saliency map), and explain why Pac-Man made that decision.

The prediction task consisted of 2 training trials and 10 test trials. Each participant was assigned to either the screenshot group or the saliency map group. Dependent variables include whether predictions were correct, and whether explanations were consistent with the saliency maps.

Results The average matching accuracy of the participants was 61.0% ($SD = 14.0\%$). A learning effect was found with participants having higher accuracy (65.5%) in the last half of the trials than the first half (56.5%) ($t(39) = 3.10, p = 0.04$). Comparing hit and false alarm rates, participants reported more "matches" when the video and image stimulus matched ($t(18) = 2.91, p < 0.001$). If the 40 participants are treated as a binary classifier and the percentage of their answers as an output score, a receiver oper-

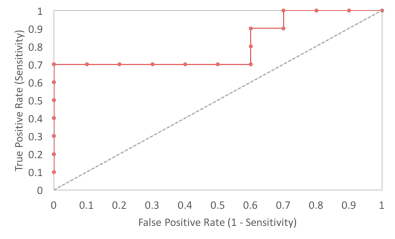


Figure 4: ROC curve of the matching task, AUC = 0.81.

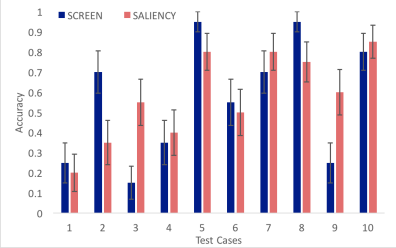


Figure 5: The mean accuracy of participants in each test cases of the prediction task. Error bars are one Standard Error from Means.

ating characteristic (ROC) curve (Fawcett, 2006) can be plotted for true positive rates versus false positive rates across a range of threshold parameters (as Fig. 4 shows). The area under the curve is 0.81 which indicates a good classification between matching and non-matching situations. In summary, human participants were able to link the object saliency maps with the game scenarios.

For the more difficult prediction task, there was no significant difference in accuracy between the object saliency map group ($58.0\% \pm 12.8\%$) and the control group($56.5\% \pm 10.4\%$). However, the main effect of trials ($F(9,342) = 11.18$, $p < 0.001$) and the interaction between trials and groups ($F(9,342) = 2.72$, $p = 0.005$) were both highly significant suggesting that characteristics of the trials had a strong influence on performance. Thus we conducted a simple effect analysis to examine differences among the 10 test scenarios (see Fig. 5). Results show that the screen-shot group has high predictive accuracy in test 2 ($p = 0.027$), while the object saliency map group has higher accuracy in tests 3 and 9 ($p = 0.007, p = 0.025$). Those three trials can help provide a deeper insight into the mechanism of how object saliency maps could help humans understand Pac-Man's learned behavior.

Trial 9 provides a good example (see Fig. 3). The Pac-Man goes down and faces a dilemma whether to turn left or keep going down. 60% participants who saw the object saliency maps predicted Pac-Man would continue going down, and objects circled and explanations focused on the dark elements or dots below. In contrast, 75% participants in the screen-shot group predicted Pac-Man would go left, and all except one of their explanations mentioned the cherry at the left side. In the scenarios generated by O-DDQN, the Pac-Man did go down for the dots. Trial 9 is a typical case in which there are multiple influencing elements and it is hard for humans to predict Pac-Man's behavior based on information from the game screen and their own knowledge of the rules and ideas about gameplay. However, displaying object saliency enables us to directly identify those objects affecting the program's decision. In other situations when the Pac-Man may make what we judge to be suboptimal choices (e.g., the Pac-Man chose a wrong direction and was eaten by a ghost), an object saliency map could be crucial to helping users and system developers understand some of the rationale behind such behaviors and the saliency map can be used as a debugging tool.

5.2. Natural Language Explanation

To present information from object saliency maps in a form more consistent with human reasoning we developed algorithms to generate relevance-sensitive textual explanations for DRL networks. We developed focused verbal explanation models of the DRL system in which verbal explanations referred to objects expected to be most important in influencing the agent's selection of next action. Prior work in generating explanations for RL systems has been limited with direct translation of policies (Hayes & Shah, 2017) restricted to simple cases while more complex environments such as Atari games (Ehsan et al., 2017) have relied on human attributions. Our approach is intermediate basing explanation on internal information but restricting its expression to a teleological form with relations satisfying human criteria for relevance. Our initial rule-based model was constructed based on prior knowledge of the Ms. Pac-Man game and its rules and consisted of a collection of allowable expressions. Object salience and valence were used to match information encoded in a saliency map with a verbal explanation. Although the rule-based model is capable of generating reasonable explanations, it lacks the generalizability and flexibility that a neural network might provide in addressing unexpected situations. To overcome the limitations of the rule-based model, a learning model was also developed in order to : 1) be more generalizable in terms of game episodes 2) be more tolerant to input noise 3) to provide explanation for future states and actions, which could help the user predict and plan. Game image, agent position map, and object saliency map act as the input for both models. Data generated by the rule-based model was employed to *train the learning model*, which consisted of two parts: an encoder for feature extraction, and a decoder for generating the explanation in natural language using an attention mechanism. The challenge for the learning model lies in extracting distinguishable features from DRL systems, especially from images with high structural similarity, such as a game image with fixed board or fixed map, which can not be fully solved by current networks.

5.2.1. Rule-based Verbal Explanation Model

The input of the rule-based model is game images and corresponding object saliency maps. The output is a verbal explanation for the given game state. The pre-defined rules and workflow used in this model were designed based on literature and a previous user study (Iyer et al., 2018). The design of the rule-based system was based on a number of assumptions. (i) *Object priority*: Since different objects in the game give Pac-Man different rewards (e.g., being eaten by the ghost means Packamn dies, eating a ghost when it becomes edible gives high positive reward), the explanation should focus on high-value objects (positive or negative). From Pakman's rules the value of objects is ghost, edible ghost, cherry, pellet, and dot. (ii) *Attention area*: User tests confirmed that participants only consider a limited area relatively close to Pac-Man when explaining its actions. this is reasonable since the Pac-Man and ghosts move only one step at a time and the rest of the objects are static, with the exception of cherries that appear and disappear dynamically. Therefore we consider only a limited area of attention with limited number of objects for the verbal explanation. (iii) *Action accordance* The expected action of Pac-Man should be approaching beneficial objects and avoiding ghosts. However, there are situations where Pac-Man has to leave beneficial object

in order to avoid ghost or Pac-Man has to approach ghost in order to chase beneficial objects. The action of Pac-Man is divided into two classes, considering consistent or inconsistent with expectations. Class #1 means the action of Pac-Man is in accordance with the expectation. Class #2 means the action is in contrast with the expectation. (iv) *Language style* To make the explanation more natural we created sentences that describe the Pac-Man's actions (moving directions) followed by the objects that motivate the current action. Relative coordination is employed to indicate the position of objects with relation to Pac-Man. An explanation template was designed accordingly. For example, a typical sentence is "The Pac-Man moves up to eat the dot above her".

5.2.2. Learning-based Verbal Explanation Model

The learning model consists of two stages. The first stage is image processing in order to get the game image feature map. The second stage generates verbal content based on the image process result. The **Image encoder** encodes game image, Pac-Man position map, and object saliency map for the verbal decoder. The three input channels provide information from different aspects: Game image provides environment information. Pac-Man position map contains Pac-Man location information and the object saliency map provides the saliency weight of each object. The **Verbal decoder** generates a verbal description consistent with the image encoder output. A sequence generation model generates verbal explanations verbatim. An attention mechanism then selects the most salient output from the encoder. To capture dynamic game information five previous frames of the game image, five previous frames of the Pac-Man position map, and a frame of the object saliency map serve as input. A verbal explanation derived from the game image is generated as output.

5.2.3. Experimental Results

In this part, both the verbal explanation of rule-base model and learning model are quantitatively evaluated.

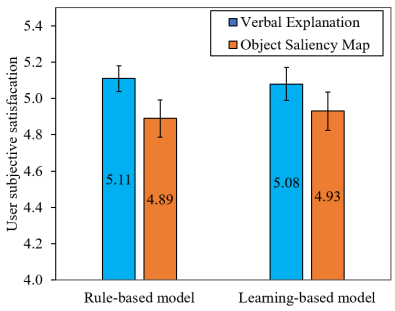


Figure 6: Subjects satisfaction score.

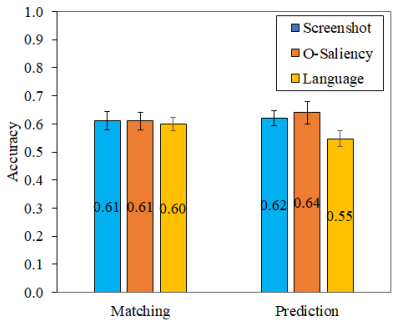


Figure 7: The mean accuracy of participants.

User tests. User tests were conducted in our laboratory and through Mechanical Turk to validate the appropriateness and effectiveness of both the rule-based and learning-based explanation generation models.

Experimental design. The matching and prediction tasks were conducted in a laboratory environment with 17 paid participants recruited from the University of Pittsburgh community. The Task settings and game episodes were identical to those used in the earlier evaluation of object-saliency maps, except some stimulus materials were replaced with natural language explanations generated by the rule-based model.

The second task to evaluate the acceptability of DNN generated explanations was conducted online using Amazon Mechanical Turk. The online questionnaire consisted of an introduction section and two evaluation tasks. The introduction contained basic background knowledge about the Ms. Pac-Man game, object saliency maps and the language generation model. On each trial participants reported their satisfaction with an explanation presented either visually (saliency map) or verbally (rule-based or learning-based). In the visual evaluation trials, a game screen-shot and the corresponding object saliency map were presented. In the verbal evaluation trials, a natural language explanation generated by either the rule-based or learning based model was given in addition to the two images. Responses were collected on a Likert scale ranging from 1 (strongly disagree) to 7 (strongly agree).

The 10 trials on each task contained randomly selected scenarios from the game episodes played by DRL. The sequence of two tasks and trials in each task was randomized to counterbalance the learning effect. For the rule-based and learning models, tests were conducted separately on different groups of participants to avoid interference. The questionnaire was deployed on Qualtircs.com for public access.

Results. The performance of matching and prediction participants was compared directly with the historical data from the previous experiment. There was no significant difference between natural language explanation and the object-saliency visualization.

For the online questionnaire, 150 samples are kept after removing abnormal data. For the rule-based model, the average rating of verbal and visual tasks were 5.11 ± 0.07 (Mean $\pm$ Standard Error) and 4.89 ± 0.09, respectively. Paired T-test showed that the rule-based verbal explanations received significant higher subjective ratings than object saliency maps, $t(74) = 2.989, p = .004$. For the learning model, a similar pattern appears suggesting that learning-based verbal explanations (5.08 ± 0.10) are better than o-saliency maps (4.93 ± 0.10) in terms of users' satisfaction, $t(74) = 2.020, p = .047$. The results are shown in figures 6 and 7.

Our results indicate that verbal explanations consistent with human preferences for teleological explanation are found more satisfactory than visual saliency maps that do not respect this preference. However, in terms of matching or prediction accuracy there were no significant differences between the natural language explanations and the saliency maps from which they were derived.

6. Conclusions

In this chapter we present issues involving trust and transparency arising in in human interactions with autonomy that uses AI and Machine Learning algorithms. Sections 1-3 review literature on trust and transparency for conventional automation in

order to extrapolate to what may be expected as we move from systems which automate a relatively narrow range of actions to autonomous agents/robots with substantially larger action spaces. We argue that as the degree of automation (extent to which system output is controlled by machine rather than human) increases, transparency of the automation usually decreases. Adopting Wickens (2018) simplified two stage (input/output) model of automation we argue that transparency (added information) at the input stage contributes more to making a system predictable than transparency at the output stage. So, a display showing the proximity of targets, for example, would be a greater help in predicting the behavior of an automated weapon than assurance or experience that the weapon is 90% effective although either might lead to a decision to rely on the automation. A convergent literature on explanation from psychology suggests that humans have a strong preference for teleological (causal) explanations and are more likely to use such models to guide their actions. From these observations we suggest that an AI/robot's ability to provide or support teleological explanations of its behavior will likely be crucial to to effective human-robot interaction and teaming.

The remainder of the chapter is devoted to examining the consequences for interacting with Deep Reinforcement Learning systems which use very high dimensional, nonlinear embeddings to generate their behavior. While DRL produces highly effective performers achieving equal or better than human performance they are opaque and often make choices baffling to humans. In a series of studies we examine mechanisms which might make DRL behavior more transparent. The first study supports transparency by making the input used by the system visible to the user in a manner similar to the automated weapon's display of target proximity. While participants were able to associate saliency maps with corresponding screen shots, their ability to predict Pac-Man's next action based on the most influential objects/regions in the display was not significantly greater than chance. A trial by trial examination shows that while saliency maps improved predictions substantially in some cases in others it depressed them.

The second set of experiments addressed the question of whether constraining explanations to teleological form could make them more effective and usable. To generate explanations, objects with high salience and their valence were matched against hypothesized rules governing Pac-Man behavior. So, for example, in a saliency map in which pellets below the Pac-Man had the highest salience and positive valence the situation might be re-expressed as, 'the Pac-Man is attracted by the pellets below'. The verbal teleological descriptions of saliency map contents performed no better than the maps themselves on the matching and prediction tasks. Because many of the collected episodes of DRL gameplay could not be translated by matching to rules, a second DRL network was trained using the matches as labeled examples. Mechanical Turk workers rated explanations generated by this DRL as satisfactory as those generated by matching to rules and felt both to be more interpretable than the saliency maps. Performance on the matching and prediction tasks, however, did not vary across conditions. These experiments suggest that while DRL networks learn to play games such as Pac-Man with a high level of skill, what they have learned and how they play may seem quite alien to a human observer. When provided with a series of screen shots or saliency maps our participants readily attributed desires such as eating pellets or avoiding ghosts to the Pac-Man yet predicting actions on this basis worked no better than chance. If we are to take advantage of the strength of DRL performance in human-robot interaction

or human-autonomy teaming this gulf between how we view problems and how they come to be solved by a learner with massive experience but none of our knowledge must be bridged. Until then we may come to trust DRL systems based on performance alone but won't be able to predict their actions or realize when they are wrong.

As the development and penetration of these systems into society increases, and as vulnerabilities of these opaque systems are identified, there is a tremendous need for (a) formulating rigorous definitions of transparency, (b) identifying dimensions of transparency and algorithms for making those dimensions operational to humans, and (c) studying their effects in human autonomy teaming. There is also an imperative to study the transparency of these systems in the broader societal context.

References

Amato, F., Felici, M., Lanzi, P., Lotti, G., Save, L., & Tedeschi, A. (2011). Trust observations in validation exercises. (pp. 216–223).

Annasamy, R., & Sycara, K. (2019). Towards better interpretability in deep q-networks. In *International Conference on Artificial Intelligence (AAAI)*. AAAI.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, .

Beck, H. P., Dzindolet, M. T., & Pierce, L. G. (2007). Automation usage decisions: Controlling intent and appraisal errors in a target detection task. *Human Factors*, *49*, 429–437.

Biros, D. P., Daly, M., & Gunsch, G. (2004). The influence of task load and automation trust on deception detection. *Group Decision and Negotiation*, *13*, 173–189.

Bitan, Y., & Meyer, J. (2007). Self-initiated and respondent actions in a simulated control task. *Ergonomics*, *50*, 763–788.

Bliss, J. P., Jeans, S. M., & Prioux, H. J. (1996). Dual-task performance as a function of individual alarm validity and alarm system reliability information. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 1237–1241). SAGE Publications Sage CA: Los Angeles, CA volume 40.

Brunelli, R. (2009). *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing.

Calhoun, G. L., Draper, M. H., & Ruff, H. A. (2009). Effect of level of automation on unmanned aerial vehicle routing task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *53*, 197–201.

Chen, J. Y., & Barnes, M. J. (2014). Human–agent teaming for multirobot control: A review of human factors issues. *IEEE Transactions on Human-Machine Systems*, *44*, 13–29.

Chen, J. Y., Procci, K., Boyce, M., Wright, J., Garcia, A., & Barnes, M. (2014). *Situation awareness-based agent transparency*. Technical Report Army research lab Aberdeen proving ground MD human research and engineering ....

Choi, E., Hewlett, D., Uszkoreit, J., Polosukhin, I., Lacoste, A., & Berant, J. (2017). Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 209–220). volume 1.

Choudlechova, A. (2016). Fair predictions aith disparate impact: A study of bias in recidivism prediction instruments. In *arXiv:1610.07524*.

Dadashi, N., Stedmon, A. W., & Pridmore, T. P. (2013). Semi-automated cctv surveillance: The effects of system confidence, system accuracy and task complexity on operator vigilance, reliance and workload. *Applied Ergonomics*, *44*, 730–738.

Davis, R. (1978). Pattern-directed inference systems. chapter Model-directed learning of production rules. New York: Academic Press.

Dixon, S., & Parasuraman, R. (2006). Automation reliability in unmanned aerial vehicle control: A reliance-compliance model of automation dependence in high workload. *Human Factors*, *48*, 474–486.

Dorsa Sadigh, A. D. D., Sastry, S., & Seshia, S. A. (2017). Active preference-based learning of reward functions. In *Robotics: Science and Systems (RSS)*.

Dzindolet, M., Peterson, S., Pomranky, R., Pierce, L., & Beck, H. (2003). The role of trust in automation reliance. *International Journal of Human-Computer Studies*, *58*, 697–718.

Dzindolet, M. T., Pierce, L. G., Beck, H. P., & Dawe, L. A. (2002). The perceived utility of human and automated aids in a visual detection task. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *44*, 79–94.

Ehsan, U., Harrison, B., Chan, L., & Riedl, M. O. (2017). Rationalization: A neural machine translation approach to generating natural language explanations. *arXiv preprint arXiv:1702.07826*, .

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recogn. Lett.*, *27*, 861–874. URL: `http://dx.doi.org/10.1016/j.patrec.2005.10.010`. doi:10.1016/j.patrec.2005.10.010.

Gao, J., & Lee, J. D. (2006). Extending the decision field theory to model operators' reliance on automation in supervisory control situations. *Trans. Sys. Man Cyber. Part A*, *36*, 943–959.

Goodman, B., & Flaxman, S. (2016). European union regulations on algorithmic decision making and "a right to explanation". In *arXiv:1606.08813*.

de Graaf, M., & Malle, B. F. (2017). How people explain action (and autonomous intelligent systems should too). In *AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction*.

Greydanus, S., Koul, A., Dodge, J., & Fern, A. (2017). Visualizing and understanding atari agents. *arXiv preprint arXiv:1711.00138*, .

Hadfield-Menell, D., Russell, S. J., Abbeel, P., & Dragan, A. (2016). Cooperative inverse reinforcement learning. In *Advances in neural information processing systems* (pp. 3909–3917).

Hancock, P. A., Deborah, B., Schaefer, K., Chen, J., de Visser, E., & Parasuraman, R. (2011). A meta analysis of factors affecting trust in human robot interaction. *Human Factors*, *53*, 517–527.

van Hasselt, H., Guez, A., & Silver, D. (2015). Deep reinforcement learning with double q-learning. *CoRR*, *abs/1509.06461*. URL: `http://arxiv.org/abs/1509.06461`.

Hayes, B., & Shah, J. A. (2017). Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction* (pp. 303–312). ACM.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., & Abbeel, P. (2017). Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, .

Huang, S. H., Held, D., Abbeel, P., & Dragan, A. D. (2019). Enabling robots to communicate their objectives. *Autonomous Robots*, *43*, 309–326.

Iyer, R., Li, Y., Li, H., Lewis, M., Sundar, R., & Sycara, K. (2018). Transparency and explanation in deep reinforcement learning neural networks. *In Proceedings of the AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society*, .

Jian, J., Bisantz, A., & Drury, C. (2000). Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, *4*, 53–71.

Jones, G., & George, J. (1998). the experience and evolution of trust: Implications for cooperation and teamwork. *Academy of Management Review*, *23*, 531–546.

Kim, B. (2015). *Interactive and interpretable machine learning models for human machine collaboration*. Ph.D. thesis Massachusetts Institute of Technology.

Kira, Z., & Potter, M. A. (2009). Exerting human control over decentralized robot swarms. In *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on* (pp. 566–571). IEEE.

Kraska, T., Beutel, A., Chi, E. H., Dean, J., & Polyzotis, N. (2018). The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data* (pp. 489–504). ACM.

Lample, G., & Chaplot, D. S. (2016). Playing fps games with deep reinforcement learning. *arXiv preprint arXiv:1609.05521*, .

Langley, P., Meadows, B., Sridharan, M., & Choi, D. (2017). Explainable agency for intelligent autonomous systems. In *AAAI* (pp. 4762–4764).

Lee, J., & Moray, N. (1992). Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, *35*.

Lee, J., & Moray, N. (1994). Trust, self confidence and operator's adaptation to automation. *International Journal of Human-Computer Studies*, *40*, 153–184.

Lee, J., & See, K. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, *46*, 50–80.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, *17*, 1334–1373.

Lewandowsky, S., Mudy, M., & Tan, G. (2000). The dynamics of trust: Comparing humans to automation. *Journal of Experimental Psychology: Applied*, *6*, 104–123.

Lewis, C. M., & Heidorn, P. B. (1991). Identifying tacit strategies in aircraft maneuvers. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*, 1560–1571.

Lewis, M. (1998). Designing for human-agent interaction. *AI Magazine*, *19*, 67.

Li, Y., Sycara, K. P., & Iyer, R. (2017). Object-sensitive deep reinforcement learning. In *GCAI 2017, 3rd Global Conference on Artificial Intelligence, Miami, FL, USA, 18-22 October 2017.* (pp. 20–35). URL: `http://www.easychair.org/publications/paper/h9zx`.

Linegang, M. P., Stoner, H. A., Patterson, M. J., Seppelt, B. D., Hoffman, J. D., Crittendon, Z. B., & Lee, J. D. (2006). Human-automation collaboration in dynamic mission planning: A challenge requiring an ecological approach. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 2482–2486). SAGE Publications Sage CA: Los Angeles, CA volume 50.

Lipron, Z. (2013). The mythos of model interpretability. In *arXiv:1606.03490v3*.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, .

Liu, C., Hamrick, J. B., Fisac, J. F., Dragan, A. D., Hedrick, J. K., Sastry, S. S., & Griffiths, T. L. (2016). Goal inference improves objective and perceived performance in human-robot collaboration. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems* (pp. 940–948). International Foundation for Autonomous Agents and Multiagent Systems.

Lombrozo, T. (2006). The structure and function of explanations. *Trends in cognitive sciences*, *10*, 464–470.

Lyons, J., & Stokes, C. (2012). Human-human reliance in the context of automation. *Human Factors*, *54*, 112–121.

Lyons, J. B. (2013). Being transparent about transparency: A model for human-robot interaction. In *2013 AAAI Spring Symposium Series*.

Lyons, J. B., & Havig, P. R. (2014). Transparency in a human-machine context: approaches for fostering shared awareness/intent. In *International Conference on Virtual, Augmented and Mixed Reality* (pp. 181–190). Springer.

Lyons, J. B., Koltai, K. S., Ho, N. T., Johnson, W. B., Smith, D. E., & Shively, R. J. (2016). Engineering trust in complex automated systems. *ergonomics in design*, *24*, 13–17.

Madsen, M., & Gregor, S. (2000). Measuring human-computer trust. In *11th australasian conference on information systems* (pp. 6–8). Citeseer volume 53.

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of management review*, *20*, 709–734.

Mercado, J. E., Rupp, M. A., Chen, J. Y., Barnes, M. J., Barber, D., & Procci, K. (2016). Intelligent agent transparency in human–agent teaming for multi-uxv management. *Human factors*, *58*, 401–415.

Meyer, J. (2004). Conceptual issues in the study of dynamic hazard warnings. *Human Factors*, *46*, 196–204.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *CoRR*, *abs/1602.01783*. URL: `http://arxiv.org/abs/1602.01783`.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, .

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015a). Human-level control through deep reinforcement learning. *Nature*, *518*, 529–533. URL: `http://dx.doi.org/10.1038/nature14236`.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015b). Human-level control through deep reinforcement learning. *Nature*, *518*, 529.

Moray, N., & Inagaki, T. (1999). Laboratory studies of trust between humans and machines in automated systems. *Transactions of the Institute of Measurement and Control*, *21*, 203–211.

Moray, N., Inagaki, T., & Ito, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, *6*, 44–58.

Morgan, B. B., Glickman, A. S., Woodard, E. A., Blaiwes, A. S., Salas, E., Campbell, W. J., Miller, D. L., Montero, R. C., & Zimmer, S. (1986). *Measurement of team behaviors in a Navy training environment*. Old Dominion University Research Foundation.

Muir, B., & Moray, N. (2013). Trust in automation: 2. experimental studies of trust and human intervention in a process control simulation. *Ergonomics*, *39*, 429–460.

Nam, C., Walker, P., Lewis, M., & Sycara, K. (2017). Predicting trust in human control of swarms via inverse reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (pp. 528–533). IEEE.

Nam, C., Walker, P., Li, H., Lewis, M., & Sycara, K. (2019). Models of trust in human control of swarms with varied levels of autonomy. *IEEE Transactions on Human-Machine Systems*, .

Ng, A. Y., Russell, S. J. et al. (2000). Algorithms for inverse reinforcement learning. In *Icml* (p. 2). volume 1.

Oh, J., Suppe, A., Duvallet, F., Boularias, A., Vinokurov, J., Navarro-Serment, L., Romero, O., Dean, R., Lebiere, C., Hebert, M., & Stentz, A. (2015). Toward mobile robots reasoning like humans. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Parasuraman, R., & Manzey, D. (2010). Compacency and bias in human use of automation:an attentional integration. *Human Factors*, *52*, 381–410.

Parasuraman, R., Sheridan, T., & C, W. (2000). A model of types and levels of human interaction with automation. *IEEE Transactions on SMC, Part A: Systems and Humans*, *30*, 286–297.

Pezzulo, G., Donnarumma, F., & Dindo, H. (2013). Human sensorimotor communication: A theory of signaling in online social interactions. *PloS one*, *8*, e79876.

Ramachandran, D., & Amir, E. (2007). Bayesian inverse reinforcement learning. In *IJCAI* (pp. 2586–2591). volume 7.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). ACM.

Riedmiller, M. (2005). Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning* (pp. 317–328). Springer.

Riley, V. A. (1994). *Human use of automation*. Ph.D. thesis University of Minneapolis.

Robinette, P., Li, W., Allen, R., Howard, A. M., & Wagner, A. R. (2016). Overtrust of robots in emergency evacuation scenarios. In *ACM/IEEE International Conference on Human-Robot Interaction* (pp. 101–108).

Rummery, G. A., & Niranjan, M. (1994). *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering.

Sadler, G., Battiste, H., Ho, N., Hoffmann, L., Johnson, W., Shively, R., Lyons, J., & Smith, D. (2016). Effects of transparency on pilot trust and agreement in the autonomous constrained flight planner. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)* (pp. 1–9). IEEE.

Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On teams, teamwork, and team performance: Discoveries and developments. *Human factors*, *50*, 540–547.

Salas, E., Sims, D. E., & Burke, C. S. (2005). Is there a "big five" in teamwork? *Small group research*, *36*, 555–599.

Schaefer, K. E. (2013). *The perception and measurement of human-robot trust*. Ph.D. thesis University of Central Florida Orlando, Florida.

Seppelt, B. D., & Lee, J. D. (2007). Making adaptive cruise control (acc) limits visible. *International journal of human-computer studies*, *65*, 192–205.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T. et al. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, .

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, .

Simpson, A., Brander, G., & Portsdown, D. (1995). Seaworthy trust: Confidence in automated data fusion. *The Human-Electronic Crew: Can we Trust the Team*, (pp. 77–81).

Stanton, N. A., Young, M. S., & Walker, G. H. (2007). The psychology of driving automation: a discussion with professor don norman. *International journal of vehicle design*, *45*, 289–306.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* volume 1. MIT press Cambridge.

Sycara, K. P., Lewis, M., Lenox, T., & Roberts, L. (1998). Calibrating trust to integrate intelligent agents into human teams. In *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on* (pp. 263–268). IEEE volume 1.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intiriguing properties of neural networks. In *arXiv:1312.6199*.

Thines, C. A. . B. G. E., G. (1991). *Michotte's experimental phenomenology of perception*. Hillsdale, NJ: Erlbaum.

Verberne, F. M., Ham, J., & Midden, C. J. (2012). Trust in smart systems: Sharing driving goals and giving information to increase trustworthiness and acceptability of smart systems in cars. *Human factors*, *54*, 799–810.

de Visser, E., Parasuraman, R., Freedy, A., Freedy, E., & Weltman, G. (2006). A comprehensive methodology for assessing human-robot team performance for use in training and simulation. In *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting* (pp. 2639–2643). HFES.

de Visser, E. J., Cohen, M., Freedy, A., & Parasuraman, R. (2014). A design methodology for trust cue calibration in cognitive agents. In *International Conference on Virtual, Augmented and Mixed Reality* (pp. 251–262). Springer.

Wang, N., Pynadath, D. V., & Hill, S. G. (2016). Trust calibration within a human-robot team: Comparing automatically generated explanations. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction* (pp. 109–116). IEEE Press.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, .

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. Ph.D. thesis University of Cambridge England.

Wickens, C. (2018). Automation stages & levels, 20 years after. *Journal of Cognitive Engineering and Decision Making*, *12*, 35–41.

Williams, J. D., Asadi, K., & Zweig, G. (2017). Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*, .

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*, 229–256.

Wright, J. L., Chen, J. Y., Barnes, M. J., & Hancock, P. A. (2016). The effect of agent reasoning transparency on automation bias: An analysis of response performance. In *International Conference on Virtual, Augmented and Mixed Reality* (pp. 465–477). Springer.

Xu, A., & Dudek, G. (2015). Optimo: Online probabilistic trust inference model for asymmetric human-robot collaborations. In *Conference on Human Robot Interaction* (pp. 221–228). ACM.

Zahavy, T., Ben-Zrihem, N., & Mannor, S. (2016). Graying the black box: Understanding dqns. In *International Conference on Machine Learning* (pp. 1899–1908).

Zhang, A., Ballas, N., & Pineau, J. (2018a). A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937*, .

Zhang, C., Vinyals, O., Munos, R., & Bengio, S. (2018b). A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*, .

Zhou, A., Hadfield-Menell, D., Nagabandi, A., & Dragan, A. D. (2017). Expressive robot motion timing. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (pp. 22–31). ACM.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., & Farhadi, A. (2016). Target-driven visual navigation in indoor scenes using deep reinforcement learning. *arXiv preprint arXiv:1609.05143*, .