Deep Reinforcement Learning of Abstract Reasoning from Demonstrations

Madison Clark-Turner
University of New Hampshire
Durham, New Hampshire
mbc2004@cs.unh.edu

Momotaz Begum
University of New Hampshire
Durham, New Hampshire
mbegum@cs.unh.edu

ABSTRACT

Extracting a set of generalizable rules that govern the dynamics of complex, high-level interactions between humans based only on observations is a high-level cognitive ability. Mastery of this skill marks a significant milestone in the human developmental process. A key challenge in designing such an ability in autonomous robots is discovering the relationships among discriminatory features. Identifying features in natural scenes that are representative of a particular event or interaction (i.e. 'discriminatory features') and then discovering the relationships (e.g., temporal/spatial/spatio-temporal/causal) among those features in the form of generalized rules are non-trivial problems. They often appear as a 'chicken-and-egg' dilemma. This paper proposes an end-to-end learning framework to tackle these two problems in the context of learning generalized, high-level rules of human interactions from structured demonstrations. We employed our proposed deep reinforcement learning framework to learn a set of rules that govern a behavioral intervention session between two agents based on observations of several instances of the session. We also tested the accuracy of our framework with human subjects in diverse situations.

CCS CONCEPTS

• **Theory of computation → Automated reasoning**; **Abstraction**; • **Computing methodologies → Knowledge representation and reasoning**; *Cognitive robotics*; Vision for robotics; Supervised learning;

KEYWORDS

Abstract Reasoning, Learning from Demonstration, Deep Learning

ACM Reference Format:
Madison Clark-Turner and Momotaz Begum. 2018. Deep Reinforcement Learning of Abstract Reasoning from Demonstrations . In *HRI '18: 2018 ACM/IEEE International Conference on Human-Robot Interaction, March 5–8, 2018, Chicago, IL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3171221.3171289

HRI '18, March 5–8, 2018, Chicago, IL, USA

ACM ISBN 978-1-4503-4953-6/18/03…$15.00
https://doi.org/10.1145/3171221.3171289

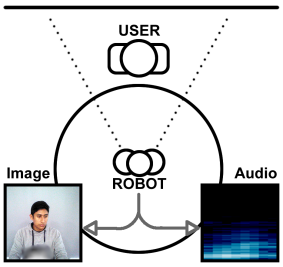


Figure 1: The Testing Environment

1 INTRODUCTION

Humans learn to perform many tasks from observing demonstrations given by others. The goal of Learning from Demonstration (LfD) research in robotics is to develop a generalizable policy for performing a task based on demonstrations delivered by lay users [3, 9]. While significant progress has been made in emulating low-level tasks that involve the understanding of motion trajectories [2, 7, 26, 28], comparatively little focus has been directed towards applying LfD methodologies to learn high-level tasks that involve understanding high-level reasoning of humans [8, 12].

Our everyday life is full of interactions with fellow humans and common objects. A majority of these interactions follow rules set by social norms or by the goal of the interaction. A human can closely approximate such rules merely from observing several instances of the interaction. This is an ability, commonly known as abstract (nonverbal) reasoning in cognitive psychology [19], that children start to develop at a later stage in the developmental process and fully master before they reach adulthood. Artificial design of abstract reasoning in robots is a highly diverse research field standing at the cross-section of cognitive science, AI, computer vision, human activity recognition, and robotics. In this paper, we want to analyze this problem purely through the lens of robotics and LfD: how can a robot identify the rules that govern a set of interactions between humans or humans and objects from observations and apply them in a similar situation? Object recognition/feature identification and symbol grounding [17] are two integral components of this problem. In contemporary LfD literature these two problems are either solved along with the original LfD problem [12] or simplifying assumptions are made about discriminatory features, objects, and symbols [8, 14]. In this paper, however, we adopt a different approach and investigate the potential of using raw perceptual data to directly learn the underlying rules of high-level events through contemporary reinforcement learning techniques.

Deep reinforcement learning (DRL) has proven to be a powerful tool for extracting insights from image data and has seen significant use at playing games [15, 23, 25]. Compared to video game playing, LfD use-cases in robotics offer a more challenging environment for DRL since images, in this case, represent natural environments and interactions which are inherently subjected to uncertainties. Deep convolutional neural networks (CNN) are able to identify concurrent, discriminatory features that occur in multiple training examples [24]. This ability is highly desirable for LfD algorithms, especially those that extract policies from image data. For example, a recent work in [30] used CNNs to generate a reward function from video training data in order to teach a robot through a separate reinforcement learning algorithm. In this research we aim to integrate feature extraction and policy learning for high-level LfD within the same framework. The proposed DRL model is capable of deriving a set of generalized rules of interaction between two agents based on several demonstrations of the interaction. We created a real use-case from an interaction between two agents designed on the principles of Applied Behavior Analysis (ABA) [4]. ABA is a set of highly reputed methodologies for structuring interventions for improving socially meaningful human behaviors. ABA has been successfully used to teach social skills to children with autism spectrum disorder (ASD) [16]. A previous work used ABA to deliver robot-mediated behavioral interventions to children with ASD [5] through tele-operation. In this paper we tested the power of our DRL framework to autonomously derive the rules of interactions of an ABA-based intervention from observing multiple instances of the same intervention. The major contribution of this paper is a novel methodology for learning abstract reasoning from demonstrations data. Aside from that, the proposed framework shows immense potential to automate the process of robot-mediated autism intervention, a domain largely dominated by tele-operated robots [6]. We evaluated our system with human subjects in diverse cases.

2 BACKGROUND

Learning high level concepts and human-style reasoning from demonstrations is a relatively under-explored domain in contemporary LfD research. The handful of works in this area, possess a strong focus on understanding the spatial reasoning in observed events. For example, the concept of sorting/stacking objects based on color and shape was grounded in [12]. A demonstrator performed a series of manipulation tasks and the robot inferred the task goal (of stacking objects in a certain manner) by analyzing the visual features of the manipulated objects in a conceptual space. The work in [22] took a different approach and used a Gaussian process classifier to teach a similar spatial relation while abstracting some visual properties of the objects away. The final goal of a series of manipulation tasks (e.g. pick-n-place operations) was inferred from observations in [14], where abstraction of a task goal is achieved through the design of a symbolic planner that ensures the robot reaches the demonstrated setting corresponding to the goal, irrespective of initial conditions. For example, while setting a dinner table, such a planner will ensure that the plate for the main dish is always located under the soup plate, but the order in which the plates were placed on the table does not matter. A conceptually similar task goal was taught to a robot in [8] through a task recipe (a collection of abstract concepts related to objects and locations involving a task). Location and object invariance was achieved by training a classifier for each concept with data from demonstrations of the same task with different objects and from different initial orientations. In these works, either perception was considered as a black box or simplifying assumptions were made for detection of discriminatory features, e.g. by hand-picking of features or restricting the complexity of the environment.

Human activity recognition (HAR) is a domain outside of robotics which has made significant progress in understanding high-level reasoning from observed events. HAR research typically analyzes human movements to infer different daily activities in video data, e.g. walking, cooking, drinking, etc. Consequently, various probabilistic graphical models have been developed to explain temporal/spatio-temporal relationships among perceived visual events [1]. A common practice among all graphical model-based HAR research is for hand-picked visual and temporal features to define the state space (or nodes) and, in turn, state transition probabilities [1]. Even a sophisticated, well-trained model may fail if feature detection does not work. Deep learning relieved the burden of hand-picking the best features and has been quickly adopted in HAR research, triggering the recent trend of end-to-end learning, recognition, and labeling of sequential tasks in video data [13, 20, 21]. Learning of abstract reasoning from data (i.e. high-level LfD), however, is subtly different than video labeling through deep learning. Video labeling generates a monolithic 'semantic explanation' of the perception, while high-level LfD requires understanding of the diminutive components, both objects and rules, that lead to that same explanation, with the hope of re-using those components to explain a novel perception. To this end we propose the deep reinforcement learning framework that combines the feature abstraction ability of CNNs with Q-learning to understand high-level reasoning from raw image data.

3 DEEP Q-NETWORK

The Deep Q-Network (DQN) is a popular model for generating policies in reinforcement learning problems [25]. DQNs, extensions of typical Q-learning, use pixel representations (images) of the state space as input and then output the estimated value associated with each of the available actions (q-values). These values are generated from CNN filters which, after training, are capable of extracting and assigning significance to features in an image. Inputs to the DQN take the form of tuples $< s, a, r, s', a' >$ which include the observations and actions performed in both the current and subsequent states (s, a and s', a' respectively) along with the most recent reward (r).

Unlike traditional classification methods that employ CNNs, the DQN is represented using two identical networks: a primary network that is updated each iteration (Q) and a secondary network that is updated infrequently ($\hat{Q}$). The $\hat{Q}$ network is used to generate an estimate of the reward in the subsequent state. This estimate is used to generate more accurate q-values from Q. The expected reward that is used when optimizing Q is

$$y_i = \begin{cases} r_t, & \text{if } s' \text{ is terminal} \\ r_t + \gamma \max_a \hat{Q}(s', a; \theta), & \text{otherwise} \end{cases} \quad (1)$$

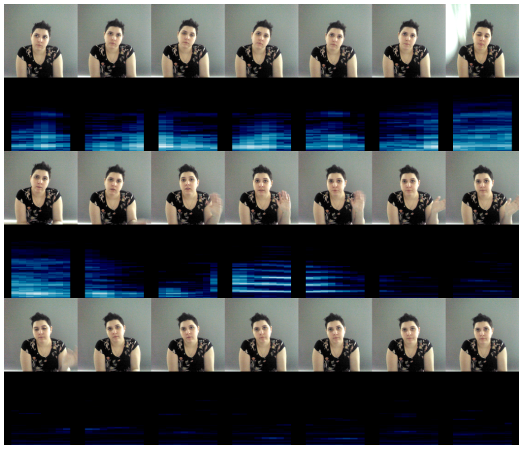


Figure 2: Example Training Data. Data presented shows the RGB and the audio channels. Frames have been removed to make the presentation more concise.

where a discount factor (γ) is used to bias the system towards receiving rewards sooner or later. The weight and bias values, θ, in $\hat{Q}$ are updated infrequently in order to prevent the q-values in Q from diverging.

Optimization of the DQN occurs through the minimization of the loss function (L)

$$L = (y_i - \max_a Q(s, a; \theta))^2 \quad (2)$$

where y_i' is the DQN's predicted q-values. The subsequent action selected for the subsequent time step is

$$a = argmax_a Q(s, a; \theta) \quad (3)$$

In typical reinforcement settings, classifiers are trained using sequential examples as simulations explore the problem's state space. DQNs introduce experience replay to randomize the order in which this data is fed to the network in order to prevent the network's output from converging or diverging as a result of of the influence of sequential actions [25]. Our model learns from examples collected offline, as opposed to receiving inputs while a simulation is running. As a result we deliver our training data in a randomized order to emulate the presence of an experience replay.

The predictive potential of deep networks can be further extended by integrating long short-term memory (LSTM) [18]. LSTM units allow a network to learn sequential information from across several frames of input data by maintaining a memory. The inclusion of LSTM layers to understand sequence information is common in the deep learning community who have had success with language modeling [33], video classification [36], and activity recognition problems [32].

4 A DQN TO EXTRACT ABSTRACT KNOWLEDGE FROM DEMONSTRATIONS

We focused on a real use-case, ABA-intervention, to learn abstract reasoning from demonstration data. Interactions between two agents (e.g. a therapist and a child) in an ABA-intervention setting are highly structured in nature. We designed a DQN model to learn this structure from observing several ABA-intervention sessions.

Our specific ABA-intervention's focus was on teaching social greetings, an intervention that was previously employed to teach children with ASD through tele-operated robots [5].

The interaction between two agents (a teacher and a student) evolved in the following way while following the ABA principles: *Discriminative stimuli* → *Response* → *Prompt* (if required) → *Reward* → *End session*. Here, *Discriminative stimuli* (SD) is performed by the teacher where s/he greets the student, e.g. by saying "Hi John"/ "Hello", waving hands, etc. *Response* from the student can be correct (e.g. the student says "Hi"/ "Hello", waves hands, makes eye-contact or smiles) or incorrect (e.g. student exhibits a behavior that is not socially acceptable in response to a social greeting or the student does not show any contingent response). In the case of an incorrect response, the teacher delivers a *Prompt* (PMT), e.g. by saying "John, say hi to me". Several prompts can be given but repeated failures of the student to respond terminates the intervention. In the case of a correct response, the teacher delivers a *Reward* (REW), e.g. by saying "Great job John!". The teacher can perform *End session* (END) after a correct response or due to repeated failures to respond correctly. Such an intervention is typically repeated multiple times per day for several days before a child with a developmental delay can master the skill of responding to a greeting in a socially acceptable manner.

From an artificial learning perspective, the perceptual features associated with each step of this intervention varies from teacher to teacher and, significantly, from student to student. This procedure will change entirely for a different intervention. Therefore, hand-picking features associated with the different steps and hard-coding an entire intervention is extremely tedious, if not impossible. A DQN offers an elegant way to learn the rules that a human teacher observes to govern the dynamics of this interaction, all from raw perceptual data. Video data of various students interacting with a tele-operated robot delivering the social greeting intervention are used to train our DQN model (Fig. 2).

4.1 Demonstration Data Collection

We collected data to train the DQN model through an IRB approved user study. Participants were recruited to take the role of a student while a NAO humanoid robot, tele-operated by the first author of this paper, took the role of a teacher to conduct the ABA-based social greeting intervention.

The robot was capable of performing three different actions in response to the observation of a participant: PMT, REW, and END whose functionality matches the description at the beginning of Section 4. As part of our implementation the robot would also execute a SD to initiate the interaction, but since our intervention never performs a SD in response to an observation we exclude it from the system's action-space.

Six individuals without ASD from the University of New Hampshire (4 male, 2 female) were recruited for the collection of demonstration data. Participants performed a total of 18 interactions with the robot. In 12 of the interactions the participant complied with the robot's requests, and either ignored or refused to greet the robot

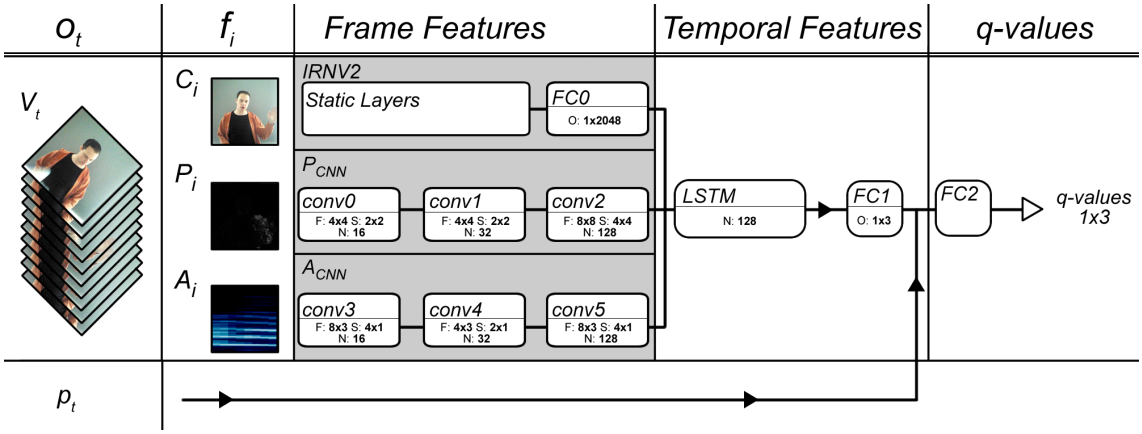


Figure 3: The Structure of our DQN. Nodes in the convolutional stacks use filter size (F), stride (S), number of filters (N), and output size (O) where present.

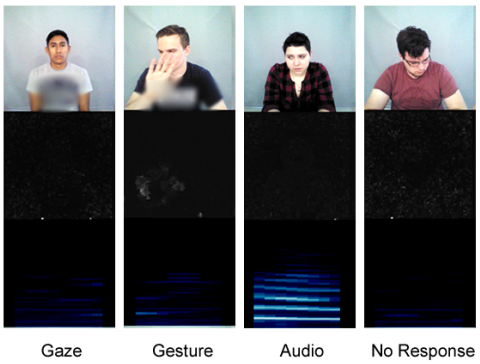


Figure 4: Various Participant Responses

in the remaining sessions. Half of the compliant interactions were delivered in order to elicit at least one prompt.

We requested that participants respond to the robot using a specific combination of

- Gaze: maintaining visual contact with the robot
- Gesture: responding to the robot's prompt with a gesture (a wave)
- Audio: responding to the robot's prompt audibly (saying "hello").

Example inputs of the different response types are depicted in Fig. 4. For the purposes of our intervention we considered responses that consisted of only gaze to be non-compliant as the participant had failed to follow the therapist's directions (to say "hello" to them).

Our complete dataset was composed of 155 videos depicting the PMT action, 118 videos depicting the REW action, and 73 videos depicting the END action. The RGB data in the REW and END actions was mirrored to increase the size of the under-represented actions. The videos varied between 106 and 184 frames in length. Fig. 1 shows a typical interaction scenario.

4.2 Data

Fig. 3 shows the structure of our DQN model. Inputs to the DQN are video data depicting the interaction between the human and the robot. Interventions (U) are sequences composed of alternating periods in which the robot delivers an action (a_t) (e.g. SD, PMT, etc.) followed by a window of explicit duration (5 seconds) in which the robot expects some response from the human participant (observation, o_t). The length of the observation window matched the delay observed during the collection of demonstrations. A typical interaction consisting of T actions by the robot is thus defined as

$$U = \langle a_0, o_0, a_1, \ldots a_t, o_t, a_{t+1}, \ldots, o_{T-1}, a_T \rangle \tag{4}$$

Each observation is a 2-tuple defined as

$$o_t = \left[V_t, p_t \right] \tag{5}$$

Here, V_t is the video data corresponding to the observation and may consist of a variable number of frames f_i, $i = 0, 1, 2, \ldots, I_t$.

$$V_t = \langle f_0, f_1, \ldots, f_{I_t} \rangle \tag{6}$$

The variable p_t in (5) is an integer representing the number of prompts that the robot has delivered prior to the current observation period. Each frame in (6) is composed of an RGB image C_i, an optical flow image P_i, and a segment of audio represented as a spectrogram (A_i). Thus,

$$f_i = \left[C_i, P_i, A_i \right] \tag{7}$$

All images and audio data were collected using the camera and microphones available on the NAO robot (Fig. 2). To generate C_i the original 640×480 image received by the NAO's main camera is cropped into a 299×299 image centered on the participant's face. The location of the face is identified using a Haarcascade filter trained on human profiles.

The cropped C_i is resized to have dimensions of 64×64 and converted to greyscale. We generate P_i using the change detection method on these images [31].

The raw audio signal, as received by the NAO's microphone, is altered to mute the first few seconds of input. This is done to remove variations in the input as a result of the robot greeting the participants by name. The signal is then subject to a combination

of spectral subtraction to remove ambient noise and smoothing through the use of an FIR filter. Finally, the signal is passed through a Mel-Spectrogram to generate a visual representation of the audio data over time [35]. The complete audio is then split to generate A_i. A total of I_t spectrograms of size 128×8, are developed from the audio signal. The split occurs with a stride of 2 so that the first three columns of A_i represent audio that took place in previous frames while the later three columns represent audio that will occur in the near future. By splitting the entire audio and combining each instance of A_i, with its corresponding C_i and P_i to generate a single frame, we hope to build strong connections between the auditory and the visual features that are observed in each frame.

4.3 Model Structure And Training

The DQN is trained to generate appropriate q-values for three actions: PMT, REW, and END. Unlike traditional reinforcement learning, DQN does not hold any explicit notion of states. Instead, all observations for a given time, o_t, are representative of the current state of the system and are passed as an input to the DQN (depicted in Figure 3) in order to generate q-values for the next action a_{t+1}. The q-values learned are based on a reward function that generates a reward of 1.0 for choosing the correct action (REW or END) that ends the session. Extending the interaction (PMT) generates a reward value of 0.2. We also maintain a discount factor 0.9. We begin by passing each frame $f_i \in V_t$ into the network sequentially. Each of the components $C_i, P_i, A_i \in f_i$ is subsequently passed into their own CNN. We use rectifier nonlinearities between each of the layers of the network and use an ADAM optimizer to train our system [29]. Our implementation is publicly available here [10].

4.3.1 Frame Features. We improve on our earlier implementation [11] by integrating a transfer learning approach when extracting features from our RGB data. Transfer learning is the use of a network, pre-trained on a large dataset, to classify novel data [27]. The low-level features that these networks have been trained to identify can be applied to discerning discriminative features in novel data. Transfer learning approaches have proven especially useful in situations where novel data is limited [30]. To that end, we pass the C_i part of our input into an InceptionResNetV2 (IRNV2) network [34]. The IRNV2 network was pre-trained on ImageNet and currently possesses the highest accuracy among ImageNet classifiers that use CNNs. We replace the final fully connected layer of the IRNV2 network with a 2048 feature vector (FC0).

The P_i and A_i inputs are also passed through their own CNNs (P_{CNN} and A_{CNN} respectively) though each are significantly smaller in scale than the IRNV2 network. Both networks conclude by generating 2048 feature vectors in order to assign equal impact to each of the three input sources. The outputs of all three CNNs (IRNV2, P_{CNN}, and A_{CNN}) are then combined to create a single 6144 feature vector. After each frame f_i of V_t has been processed the significant features from each of the I frames are placed into a $I \times 6144$ matrix which is then passed into a long-short term memory (LSTM) cell.

4.3.2 Temporal Features. In our model we use the LSTM layer to learn feature variations between frames such as the movement of a hand during a wave or the recognition of specific auditory patterns. LSTM performs this functionality by deciding whether to retain and forget features it has observed. If trained correctly, we consider this ability very useful for real world robotics applications. Much of the existing video classification and HAR work that uses deep learning relies heavily on training data that consists of only a few frames that explicitly show the task to be classified. This is rarely the case in real world systems in which the majority of a video can be composed of information that is unrelated to the classification label. In our case, the act of waving or saying 'hello' can occur over as little as 5% of a training video. LSTM has the potential to isolate the significance of a pattern in a few frames while ignoring the additional information in a sequence.

The LSTM cell also provides a secondary advantage for our system in that it can process a sequence of variable length and output a fixed length vector. Networks that fail to incorporate an LSTM layer must, instead, maintain a fixed size when feeding inputs into their networks. The demonstrations we observed can vary by up to 78 frames depending on the operators response time and system latency. Attempts to crop this video length would likely exclude important information while padding the shorter sequences to the length of the longest demonstration would introduce significant and unnecessary slow down to the system's execution time. The output of the LSTM is passed through a fully connected layer ($FC1$) in order to generate a 1×3 feature vector.

The LSTM is capable of extracting feature information between different frames of V_t but it is unable to extract the greater temporal structure of U. As an example: our system's PMT and END actions are both called after observing non-compliant responses. However, our ABA definition prohibits END from being executed before the PMT action has been called at least once. The contents of V_t could be identical (representing a non-compliant reaction), but without information about the previous actions we cannot, with certainty, correctly choose the next action. This inability to separate identical states leads us to a problem of perceptual aliasing. To resolve this confusion we provide, p_t, the number of prompts that have been delivered in a U thus far. We conclude our network by concatenating the output of $FC1$ with p_t and passing the new vector through a final fully connected layer ($FC2$) in order to generate our q-values.

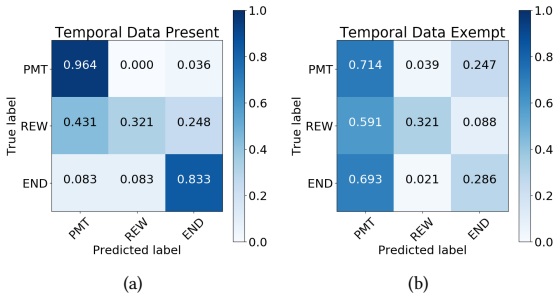
5 RESULTS

We evaluated the model described in Section 4.3 in order to assess the influence of our design choices. We used eight participants to evaluate our system, three of whom were not included in the training data. We had each of the participants perform in the same 18 interactions described in Section 4.1. Returning participants wore different attire from what was worn during the training data collection in order to ensure that the observations were novel.

5.1 Simulation

In a simulation, we compared the accuracy of a model that possessed and one that lacked the temporal information provided by p_t. The model that lacked the temporal information was identical in structure to the model depicted in Fig. 3 with the exception being that we did not concatenate p_t with the output of $FC1$ and instead passed the 1×3 output of $FC1$ directly to $FC2$.

We assessed our model's ability to correctly select the appropriate action for a given input (accuracy).



(a) (b)

Figure 5: Simulation Results

5.2 Human Participant

We evaluated our model using human participants. Based on our simulation results we decided to evaluate only the model that included the temporal information as it possessed a greater accuracy. Our on-line model begins recording the video for o_t as soon as an action is selected to be executed and has a delay of 13 seconds before the next action is selected. This provides a window of approximately 10 seconds in which a participant can respond without having the participant's response overlap the robot's action execution. Preprocessing and evaluation of o_t, collectively, takes about 1 second at the end of the observation window. During this time observations for the subsequent state are not being collected.

Responses			
Gaze	**Gestural**	**Auditory**	**Accuracy**
No	No	No	95.8%
No	No	Yes	75.0%
No	Yes	No	25.0%
No	Yes	Yes	68.8%
Yes	No	No	87.5%
Yes	No	Yes	81.3%
Yes	Yes	No	6.3%
Yes	Yes	Yes	37.5%
Total			**67.8%**

Table 1: On-line Results of Varied Responses

Out of the responses that included a gestural component 34.4% were correctly identified while 65.6% of all of the auditory responses were responded to correctly.

5.3 Sequence Analysis

We investigated our model's ability to learn sequential patterns by observing the change in q-values as we varied the duration of V_t (Fig. 6). To vary the duration we incrementally cropped the number of frames present at the end of V_t and observed the q-value predictions that were generated.

In Fig. 6 we present an analysis of four responses: an auditory response (a), a gestural response that included gaze (b), a response consisting of only gaze (c), and a response that lacked all three of the aforementioned features (d). We performed analysis after every 5 frames of input. The C_i and A_i channels show a single frame that occurred in the 5 frames prior to the evaluation.

The interactions presented are separated into three regions depending on the audio content of the video. In the first section the NAO greets the participant by name. In order to maintain the ubiquity of the videos we muted all input to the NAO's microphone while the command is given. The second section is populated by an extend period of ambient noise as the NAO executes an un-muted waving motion. Finally, there is a period in which the participant can respond without the additional noise of the robot's operation.

6 DISCUSSION

Having tested our system both in simulation and using a live model we not only show the potential of our system to learn a complex high-level reasoning skill from limited data, but also show that the model can be applied in a real-world HRI environment. We investigate the network's ability to correctly learn which actions to execute according to raw observations and the influence of our various structural considerations.

The results of both the simulation and live system indicate that the model was able to learn the desired skill and was able to execute appropriate actions for novel observations. Furthermore, the similar accuracies observed through simulation and real-world models suggests that the system can easily be adapted to a real-world scenario with little impact to the system's accuracy.

6.1 Perceptual Aliasing

When comparing the two simulated models (Fig. 5), we observed identical accuracies when selecting the REW action indicating that both models were able to learn very similar features. This was expected as the delivery of the REW action was independent of the presence of p_t.

In both the simulation and real world systems we observed a high accuracy when selecting non-compliant actions provided that the models had access to the greater temporal information about the interaction. In simulation there was a difference in accuracy of 21.0% between the model that had access to and one that lacked p_t supporting our assumption that the model would be unable to function, or would function poorly, without its inclusion.

On several occasions the PMT action was incorrectly called in response to compliant observations when $p_t = 0$. As a result the REW action observed an accuracy of 29.5% if called before the PMT action. However after the PMT action the system would correctly call the REW action 50.0% of the time and would call the END action in the remainder of the compliant responses. Our system's preference for the PMT action could be the result of obtaining a higher reward for interactions that last longer. During training the q-value generated for the PMT action vary with the values of actions performed in subsequent states. As q-values fluctuated, the PMT action may have incorrectly been associated to a value higher than it should. This inconsistency would eventually be resolved with a longer training period as the q-values become more stable.

6.2 Feature Learning

The results in Table 1 reinforce the assertion that our model was able to learn the features of the desired behavior: gestures and

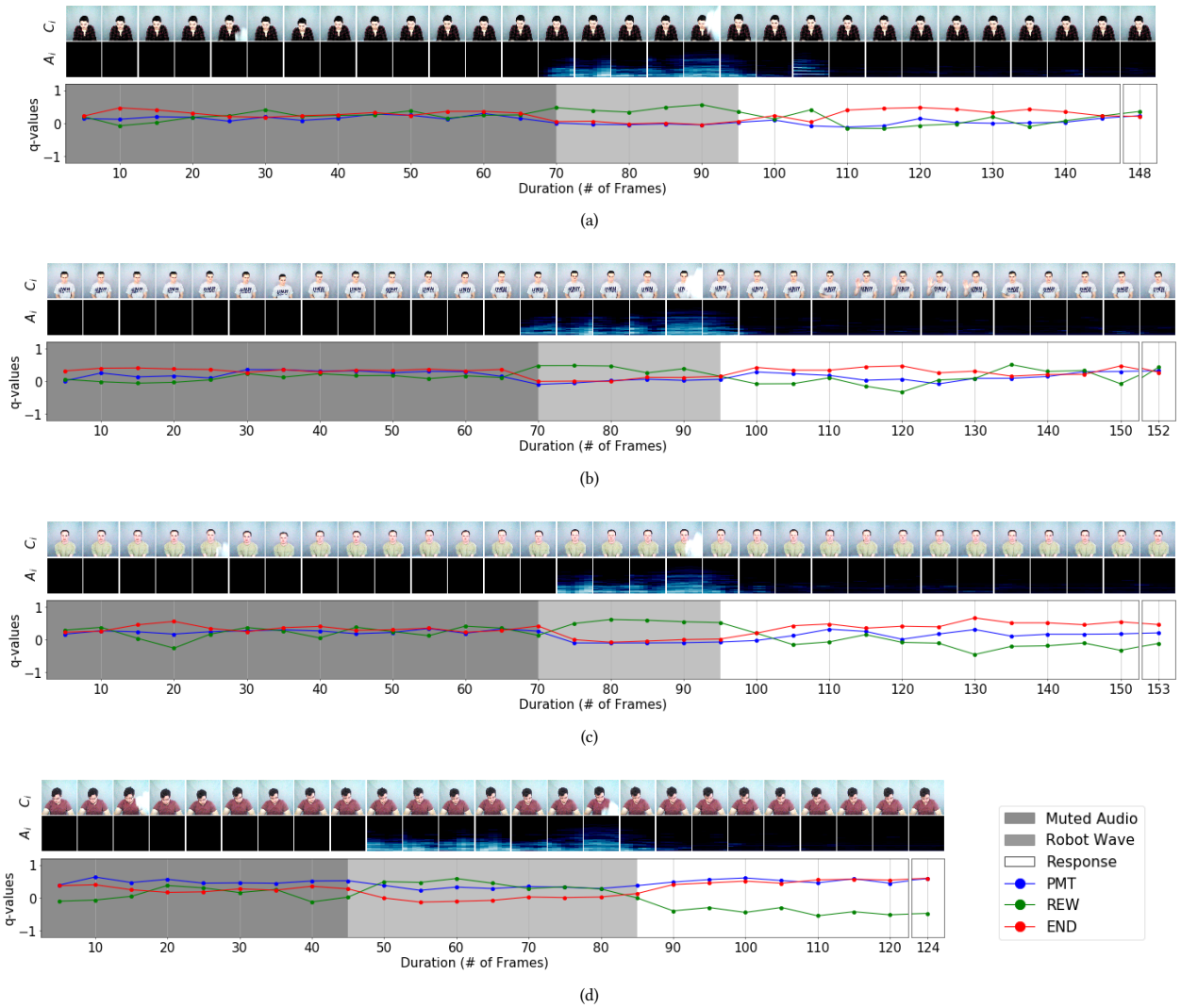


Figure 6: Q-values of Videos Segmented at Different Lengths.

auditory responses. Audio was a far more transparent symbol of compliance as opposed to gesture, which proved weaker. In all of the training examples participants had responded clearly by saying 'hello' as opposed to other auditory greetings such as 'hi'. These signals were easy to identify and were highly similar across all of the Mel-spectrograms. Gestural responses possessed greater variations. Differences such as hand position, wave speed, wave duration, and whether or not the participant kept their fingers together or apart all generated highly different representations in the optical flow. As a result our system had a harder time learning the generalizable features of a wave.

Though the accuracy of auditory signals was high, it was not 100% accurate. Instances where auditory responses failed to elicit the expected action classification could have been influenced by changes in pitch and speech duration. Furthermore, changes that occurred in C_i and P_i also influence the final action selection. This is observable in responses that required the participant to maintain visual contact with the robot. In all cases, except the auditory response, the presence of gaze was associated with a decrease in the likelihood that the model would correctly classify the action. By assigning instances where the participant maintained visual contact as non-compliant we may have inadvertently taught our model to associate gaze, regardless of other features, as a non-compliant responses.

6.3 Sequence Inference

We designed our model with an LSTM layer so that it would possess the ability to learn sequential information. We considered the identification of audio signatures and movement patterns to be pivotal to our system's performance. Our results suggest that some

useful patterns were understood but other additional features were learned that were contrary to our expectations.

We first discuss features observed across all interactions before detailing the differences that occurred when the type of response was varied. In the three instances in which prompts had already been delivered (a, b, and c) the q-values are similar during the period where the NAO's audio was muted. This indicates that there was little to no bias towards any of the actions at the start of the interaction and that bias was not introduced until after the audio was no longer muted. This was not the case for the interaction without a prompt (d) where when lacking a prompt our system maintained a bias toward non-compliant actions, specifically the PMT action. This was reinforced by our data in which the majority of compliant classifications occurred after a prompt. The muted region was followed by a section in which the system can hear the ambient noise of the NAO waving. This section, in all cases, is associated with a spike in the value of REW. Despite this fluctuation, the q-values of future frames were not altered. We can conclude that the LSTM layer had learned a pattern of silence followed by high auditory activity which is present in all of our training examples and is ignored since it is not discriminatory.

In the response that included an auditory component (Fig. 6(a)) there was a high degree of change in A_i at frame 105 when the participant responded by saying "hello". This corresponded with a sharp spike in the value of the REW action compared to PMT and END. The opposite happened during periods of quiet in which the q-values of the non-compliant actions obtained a higher value. As the video continued we saw that the q-value of REW increases and surpasses (frame 145) that of the two non-compliant actions. There is negligible difference between the C_i and A_i channels in these frames suggesting that the LSTM layer enacts this change. We conclude that our system learned that a period of quiet must follow an auditory response in order to generate the REW action assignment. While the correct action was selected in the given interaction, a different interaction occurring at a more delayed time could have been incorrectly assigned a non-compliant action and negatively affected our system's overall accuracy.

The gestural response (Fig. 6(b)) also possesses a sharp peak in the value of the REW action. This change, unlike the auditory response, did not appear at the same time as the change in A_i and instead occurred at frame 135, as the participant concluded their waving action (which had begun at frame 115). In this case, our LSTM learned that the features indicating that a wave has occurred must all be present before the q-value is altered. With the exception of frame 150, the high q-value for rewarding the participant is maintained until the response concluded.

When the observations lacked an auditory or gestural component (Fig. 6(c) and Fig. 6(d)) the non-compliant actions maintained high q-values throughout. However, there were variations between interactions that took place before and after a PMT action. In the case where PMT had been executed, the END action maintained the highest q-value during the participant's response. This advantage was large, ensuring that the other actions would not be selected. If the system had yet to perform a PMT, then the PMT action was most favored, but in this scenario the margin was small. Despite this the PMT action maintained superiority for most of the interaction. As mentioned previously, the PMT action began the interaction with a bias, but it appears that as the interaction continues this influence wanes. Furthermore, the system still has a potential, though small, to choose actions that are incorrect given the system's greater temporal structure such as performing the END action when a prompt has not been previously delivered.

7 CONCLUSIONS

High-level reasoning about human activity has persistently been a challenging task to solve. Though significant research has been performed in order to generate solutions for singular problems, few solutions have attempted to create a generalizable model. We present a system that uses deep reinforcement learning in the form of a DQN that is capable of learning skills from demonstrations. To the best of the authors' collective knowledge this is the first use case of Deep Q-Learning for high-level LfD of a human interaction that has been successfully implemented in a real world robotics application. Our model was able to learn how to perform a behavioral intervention with 68.1% accuracy in simulation and obtained comparable results when delivered via a physical robot.

An in-depth analysis of our system indicated that our model was able to correctly identify both auditory and gestural responses. Furthermore, our model correctly delivered non-compliant responses according to the greater temporal structure of the interaction. We were also able to learn several sequential features of the observations including the ability to ignore the period prior to the participant's response and the transition of features that is indicative of a gestural response.

Our model, though capable, does possess a few limitations. In order for our model to operate correctly, we had to explicitly provide information about the interaction's greater temporal structure, violating the black-box mentality of LfD. And though we were able to respond to compliant responses, our model struggled to generalize responses that exhibited greater variability in their representation. Finally, though our model was able to learn useful sequential patterns about the interaction it also learned patterns that, while correct in our dataset, were inappropriate for following the intervention's protocol.

A larger set of training demonstrations is likely to further improve our model's accuracy, but the results we obtained with only a limited sample of human interactions is indicative of the DQN's future potential. We intend to further improve upon our model by incorporating temporal features into our action selection, that don't violate the tenants of LfD, using probabilistic models such as Dynamic Bayes Networks. The internal temporal modeling of the interaction will similarly be improved by introducing video segmentation techniques to improve the LSTM nodes' focus. Future investigations will also test the potential of our design to generalize to other problems with datasets that show greater variation.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (IIS 1664554).

REFERENCES

[1] Jake K Aggarwal and Michael S Ryoo. 2011. Human activity analysis: A review. *ACM Computing Surveys (CSUR)* 43, 3 (2011), 16.

[2] S Reza Ahmadzadeh, Roshni Kaushik, and Sonia Chernova. 2016. Trajectory learning from demonstration with canal surfaces: A parameter-free approach. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 544–549.

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

[4] Donald M Baer, Montrose M Wolf, and Todd R Risley. 1987. Some still-current dimensions of applied behavior analysis. *Journal of Applied Behavior Analysis* 20, 4 (1987), 313–327.

[5] Momotaz Begum, Richard W Serna, David Kontak, Jordan Allspaw, James Kuczynski, Holly A Yanco, and Jacob Suarez. 2015. Measuring the Efficacy of Robots in Autism Therapy: How Informative are Standard HRI Metrics'. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 335–342.

[6] Momotaz Begum, Richard W Serna, and Holly A Yanco. 2016. Are robots ready to deliver autism interventions? a comprehensive review. *International Journal of Social Robotics* 8, 2 (2016), 157–181.

[7] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. 2008. Robot programming by demonstration. In *Springer handbook of robotics*. Springer, 1371–1394.

[8] Kalesha Bullard, Baris Akgun, Sonia Chernova, and Andrea L Thomaz. 2016. Grounding action parameters from demonstration. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*. IEEE, 253–260.

[9] Sonia Chernova and Andrea L Thomaz. 2014. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8, 3 (2014), 1–121.

[10] Madison Clark-Turner. 2017. Deep Reinforcement Abstract LfD. (2017). https://github.com/AssistiveRoboticsUNH/deep_reinforcement_abstract_lfd

[11] Madison Clark-Turner and Momotaz Begum. 2017. Deep Recurrent Q-Learning of Behavioral Intervention Delivery by a Robot from Demonstration Data. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*. IEEE, 1024–1029.

[12] Richard Cubek, Wolfgang Ertel, and Günther Palm. 2015. High-level learning from demonstration with conceptual spaces and subspace clustering. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2592–2597.

[13] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2625–2634.

[14] Staffan Ekvall and Danica Kragic. 2008. Robot learning from demonstration: a task-level planning approach. *International Journal of Advanced Robotic Systems* 5, 3 (2008), 33.

[15] Vlad Firoiu, William F Whitney, and Joshua B Tenenbaum. 2017. Beating the World's Best at Super Smash Bros. with Deep Reinforcement Learning. *arXiv preprint arXiv:1702.06230* (2017).

[16] Richard M Foxx. 2008. Applied behavior analysis treatment of autism: The state of the art. *Child and adolescent psychiatric clinics of North America* 17, 4 (2008), 821–834.

[17] Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42, 1-3 (1990), 335–346.

[18] Matthew Hausknecht and Peter Stone. 2015. Deep recurrent q-learning for partially observable mdps. *arXiv preprint arXiv:1507.06527* (2015).

[19] John L Horn and Raymond B Cattell. 1966. Refinement and test of the theory of fluid and crystallized general intelligences. *Journal of educational psychology* 57, 5 (1966), 253.

[20] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. 2016. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5308–5317.

[21] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[22] Johannes Kulick, Marc Toussaint, Tobias Lang, and Manuel Lopes. 2013. Active Learning for Teaching a Robot Grounded Relational Symbols.. In *IJCAI*. 1451–1457.

[23] Guillaume Lample and Devendra Singh Chaplot. 2017. Playing FPS Games with Deep Reinforcement Learning.. In *AAAI*. 2140–2146.

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[25] 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. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[26] Scott Niekum, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G Barto. 2015. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research* 34, 2 (2015), 131–157.

[27] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[28] Chris Paxton, Felix Jonathan, Marin Kobilarov, and Gregory D Hager. 2016. Do what i want, not what i did: Imitation of skills by planning sequences of actions. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 3778–3785.

[29] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithmst. (2016). http://sebastianruder.com/optimizing-gradient-descent/index.html#gradientdescentvariants

[30] Pierre Sermanet, Kelvin Xu, and Sergey Levine. 2016. Unsupervised perceptual rewards for imitation learning. *arXiv preprint arXiv:1612.06699* (2016).

[31] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*. 568–576.

[32] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. 2016. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1961–1970.

[33] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM Neural Networks for Language Modeling.. In *Interspeech*. 194–197.

[34] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.. In *AAAI*. 4278–4284.

[35] Li-Chia Yang, Szu-Yu Chou, Jen-Yu Liu, Yi-Hsuan Yang, and Yi-An Chen. 2017. Revisiting the problem of audio-based hit song prediction using convolutional neural networks. *arXiv preprint arXiv:1704.01280* (2017).

[36] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4694–4702.