2017 26th IEEE International Symposium on
Robot and Human Interactive Communication (RO-MAN)
Lisbon, Portugal, Aug 28 - Sept 1, 2017.

Deep Recurrent Q-Learning of Behavioral Intervention Delivery by a Robot from Demonstration Data

Madison Clark-Turner[1] and Momotaz Begum[1]

Abstract— **We present a learning from demonstration (LfD) framework that uses a deep recurrent Q-network (DRQN) to learn how to deliver a behavioral intervention (BI) from demonstrations performed by a human. The trained DRQN enables a robot to deliver a similar BI in an autonomous manner. BIs are highly structured procedures wherein children with developmental delays/disorders (e.g. autism, ADHD, etc.) are trained to perform new behaviors and life-skills. Mounting anecdotal evidence from human-robot interaction (HRI) research has shown that BI benefits from the use of robots as a delivery tool. Most of the HRI research on robot-based intervention relies on tele-operated robots. However, the need for autonomy has become increasingly evident, especially when it comes to the real-world deployment of these systems. The few studies that have used autonomy in robot-based BI relied on hand-picked features of the environment in order to trigger correct robot actions. Additionally, none of these automated architectures attempted to learn the BI from human demonstrations, though this appears to be the most natural way of learning. This paper represents the first attempt to design a robot that uses LfD to learn BI. We generate a model then correctly predict appropriate actions with greater than 80% accuracy. To the best of our knowledge, this is the first attempt to employ DRQN within an LfD framework to learn high level reasoning embedded in human actions and behaviors simply from observations.**

I. INTRODUCTION

Designing robots to serve in the health-care industry is considered a realistic solution to compensate for the deficit in skilled health-care professionals [1]. Recent research has supported this by indicating the clinical utility of robots in different therapeutic interventions [2]. Specifically, socially assistive robots have proven to be a potential avenue for delivering behavioral intervention (BI) to children with various intellectual and developmental disorders (IDD) [3]. Contemporary research on robot-mediated interventions (RMI), however, has relied exclusively on tele-operated robots. While tele-operation is a reasonable approach for investigating HRI factors and clinical utility, the lack of autonomy is increasingly being considered as a critical factor that is hindering the real-world deployment of robots [4]. The majority of the works that use autonomous RMI architecture hard-code the interaction. The robot's perception is limited to hand-picked features of the environment in order to trigger appropriate robot actions [4], [5]. Each child with IDD is unique, and different children may respond to the same intervention in different ways. Personalizing every intervention for each child is, therefore, highly inefficient, if not impossible. The inherent dependency on the robot-programmer also makes the real-world deployment even more challenging.

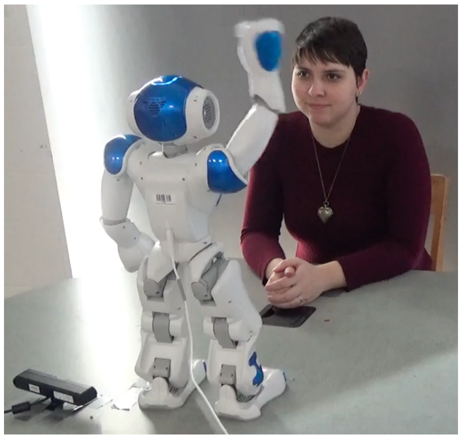
Fig. 1. The NAO humanoid robot engaged in a behavioral intervention. A depth camera is located to left of the robot.

Learning from demonstration (LfD), a popular robot learning paradigm, offers an elegant solution to these problems. LfD advocates the idea of enabling lay users to teach robots new tasks/skills simply by showing how to conduct the task and without requiring any special knowledge about robots or programming [6]. The core challenge of LfD lies in deriving a mapping between perceived features and actions (a.k.a. a policy) from a limited number of demonstrations. To date, learning generalized representations of motion trajectories is one of the most successful domains in LfD research [7]. Compared to trajectory learning, learning of high level reasoning and the spatio-temporal relationship among discrete events in everyday life from demonstration data is a relatively under-explored domain in the LfD literature. Only a handful of recent works have been dedicated to understanding simple concepts from observations, e.g., symbol grounding [8], [9], invariance in spatial relationships during a motor task [10], [11], task network structure [12], etc. Robot-based BI offers a novel application of LfD where a robot will have to learn the relationships among discrete events that are triggered by humans in a highly-structured fashion. Learning this relationship involves understanding the policy that a human therapist maintains while delivering an intervention. Such a policy is, typically, specific to the child and the intervention and evolves over time as a child makes gradual progress in skill learning. Accordingly, a LfD framework involves observing multiple BI sessions in order to learn a BI. Observations, in this case, may range from obvious responses related to the

[1]Authors are with the Assistive Robotics Lab, University of New Hampshire, Durham, NH 03824, USA mbc2004@cs.unh.edu, mbegum@cs.unh.edu

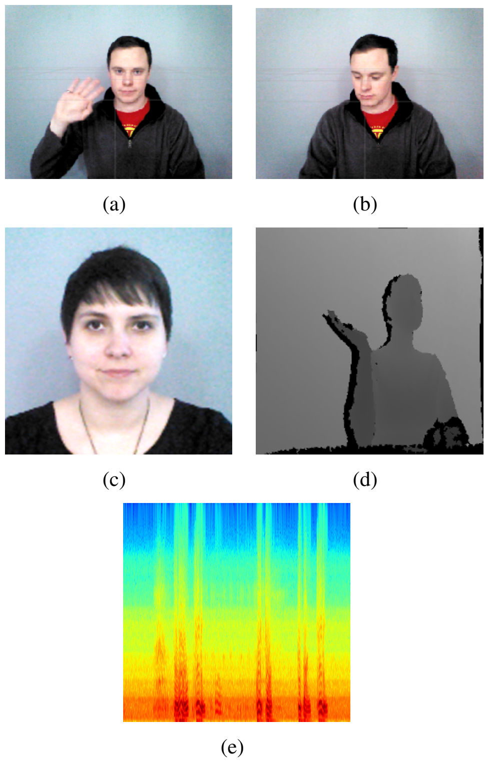
Fig. 2. Examples of compliant (a) and non-compliant (b) responses. In the compliant case the participant is looking at the robot and responding with both a wave and a vocal greeting. In the non-compliant case the participant refuses to respond to the robot and their gaze is directed away from the robot. (c) The video cropped to focus on a participants face. (d) Point cloud data observed through the depth camera as the subject waves. (e) A spectrogram of collected audio.

task/skill being taught to subtle multimodal cues expressing compliant/non-compliant behaviors of a child. Hand-crafting multimodal features related to these responses and cues for various interventions is an extremely tedious task. But the correct action of a robot (and of a human therapist, too) always depends on identifying the appropriate features. The recent development of the Deep Q-Network (DQN) offers an elegant form of human-style decision making without worrying about low-level feature selection [13] .

In this paper, we present an LfD framework capable of learning the steps in a BI procedure simply from observing multiple sessions of that procedure. We use a variant of DQN, the deep recurrent Q-network (DRQN), which is capable of learning feature information observed in pixel data received over several time steps. DQNs are a form of reinforcement learning that has received significant attention since it was first used to develop policies that could beat human players at several Atari video games [13]. The DQN has since been applied to several similar problems including the board game Hex [14], the video game Super Smash Bros.[15], and language understanding in text-based games [16] and it has proven to be a reliable method for policy learning in problems with large state spaces. Despite these apparent merits, DQN has, until now, failed to find application in LfD problems and has been poorly represented in robotics domains in general. Delivery of a BI involves complex human reasoning which is manifested as a highly-structured interaction. This makes DRQN an elegant choice for learning BI from observation data.

To the best of our knowledge, this paper makes the first effort to employ a DRQN for learning relationships among human-triggered discrete events from demonstration data. This is also the first learning-based approach for robot-mediated behavioral intervention.

This paper is structured as follows: Section II provides a description of the BI and the DRQN, which is used to learn an appropriate policy. Section III describes the methods used to collect our training data and the procedure used to train our DRQN. Section IV provides an evaluation of our system trained on the demonstration data. Section V summarizes our findings.

II. DRQN FOR BEHAVIORAL INTERVENTION

In this work we adopted applied behavior analysis (ABA), a popular and successful method for designing BI for children with IDD [17]. Our previous tele-operated RMI study was able to teach a group of autistic children a new behavior with a 20% success rate using an ABA-based BI [18]. Accordingly, we adopted the same ABA-based BI protocol for the work presented in this paper but, instead of tele-operation as in [18], we now learn this protocol from observations.

A. A social greeting BI

An example of the social greeting BI that we learned is shown in Table I. In a typical ABA-based BI session the therapist begins by delivering a verbal command and then awaits a response from their client. The client can provide either a compliant or non-compliant response. In the compliant case the therapist provides some form of reinforcement such as verbal praise. If the client is noncompliant or responds incorrectly then the therapist can try to deliver a corrective prompt encouraging the client to provide the compliant response. Several corrective responses can be delivered until either a compliant response is observed or the therapist intentionally ends the session. Reinforcement and prompts must be provided at specific times in order to properly encourage the desired behaviors. Failure to execute the intervention in the structured manner can negatively impact the client's ability to learn the desired skill.

B. Modeling the social greeting BI for robot training

We train the robot to perform four actions: *command*, *prompt*, *reward*, and *abort*. The *command* and *prompt* actions cause the robot to wave and greet the participant. The later delivers a corrective prompt that describes what is expected of the participant: that the user respond by saying "hello". Both *reward* and *abort* are actions that transition the problem to terminal states. The *reward* action is functionally equivalent to performing both the "User response is positive" branch of step 3 in Table I followed by step 4, rewarding the subject for responding and then ending the session. The *abort*

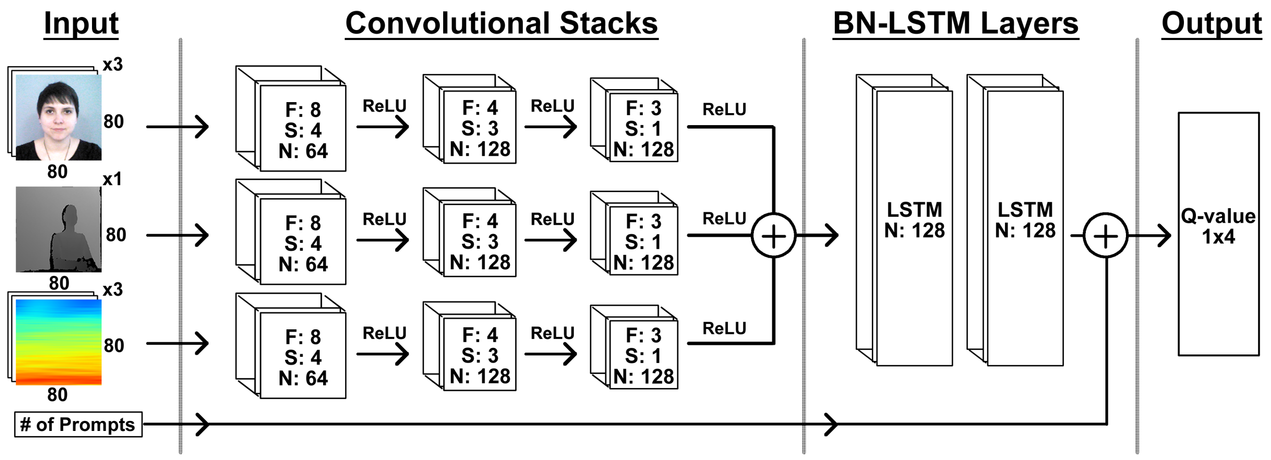

Fig. 3. The structure of the DRQN for behavioral intervention. Nodes in the convolutional stacks use filter size (F), stride (S), and number of filters (N).

#	Speaker	Behaviors
1	Therapist (Robot)	Wave hand as shown in Figure 1 and say “Hello, X”.
2	User X	[positive response] User responds by saying “Hello” or by waving. [negative response] User does not respond to the robot or responds incorrectly.
3	Therapist (Robot)	[if User response is positive] say “Great Job!” then go to step 4. [if User response is negative] Wave hand and prompt “X, say hello to me.” Go to step 2. [if User response is negative consistently] Go to step 4.
4	Therapist (Robot)	Say “Good Bye, X.”

TABLE I
EXAMPLE ABA INTERACTION

action follows the “User response is negative consistently” branch of step 3 and concludes the session.

After the *command* and *prompt* actions are performed our model takes observations of the participant in the form of video, depth, and audio data. Collectively we consider these observations as the state for our problem.

We provide positive rewards when actions are performed correctly or no reward when they are performed incorrectly. We also maintain a discount value (γ) of 0.9 to bias policies towards obtaining rewards sooner as opposed to later.

C. Deep Recurrent Q-Learning of BI

Q-learning is a common reinforcement learning approach that generates a policy by matching state(s)-action(a) combinations to Q-values ($Q(s,a)$). Q-values are obtained from repeated passes through the world space and indicate the expected value of performing the given action while in the provided state. Successfully reaching a terminal state updates the Q-values of all state-action pairs that were observed by the agent by using

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a\prime} Q(s\prime, a\prime) - Q(s,a)) \quad (1)$$

Traditional Q-learning is a reasonable solution to small problems provided that there exists the expectation that most if not all of the state-action pairs will be investigated. Unfortunately, many real-world problems do not possess this luxury and in cases where the state space is expansive, such as our problem which uses video to represent the state, predictive models must be used to estimate the Q-values for a given state-action pair.

The deep Q-network (DQN) is one such model that uses convolutional neural networks (CNN) to assign Q-values to actions given the current state and a combination of weights and biases (written together as θ). The DQN approaches a problem that is fundamentally different from CNNs in that it involves interactions that persist over several time steps. As a result the DQN model (Q) has two structural differences not found in CNNs: experience replay and parameter setting using a second model ($\hat{Q}$). Experience replay refers to the the use of a replay memory (D) that stores experience tuples of the form $e_t = (s_t, a_t, r_t, s_{t+1})$. Experience tuples are examples from a training set that are composed of the current state (s_t), the action that was performed in that state (a_t), the reward that was received after performing that state-action combination (r_t), and the state that the system subsequently transitioned to (s_{t+1}). In our problem our state is represented by a video stream with accompanying audio and depth information along with the number of prompts delivered so far. Our available actions and the rewards associated when performing them are listed in Section II-B. The DQN is trained by randomly selecting tuples from D using s_t as the input and using a one-hot encoding of a_t multiplied by r_t for labels. We use a structurally identical model $\hat{Q}$ to generate the expected reward for the subsequent state. The loss function

$$L_i(\theta_i) = E_{(s_t,s_t,r_t,s_{t+1})}[(y_i - Q(s_t, a_t; \theta_i))^2] \quad (2)$$

is used to update the i-th iteration's values for θ. In the case where s_t is a non-terminal state

$$y_i = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a; \theta^-) \quad (3)$$

otherwise

$$y_i = r_t \tag{4}$$

The values (θ) of $\hat{Q}$ are infrequently updated to be the same as those in Q. By delaying updates to $\hat{Q}$ the updates in Q become more stable, and we prevent oscillation in the training data [19].

When obtaining observations in our BI we accumulate information over an extended and variable period of time. It is inappropriate to assume that participants will begin responding at the same time and that their responses will be a uniform length, especially among IwASD. As such we use an extension of the DQN, the deep recurrent Q-network (DRQN), to incorporate time sensitive and sequential information into our predictions. DRQNs alter the DQN by replacing the final fully connected layer of the DQN model with a long-short term memory (LSTM) layer [19]. LSTM provides the DQN with a memory allowing new input data to influence the Q-values with the context of previous data. LSTM cells have been used in many recurrent neural network (RNN) applications including language modeling [20] and video classification [21]. The LSTM cell is represented using the tuple (c, f, i, o, h) [22]. Where c, f, i, and o refer to the *cell*, *forget gate*, *input gate*, and *output gate* vectors all of which are h in length. The *cell* contains data that persists across the entirety of the sequence being read into the RNN. The *cell* acts as the memory for the RNN and is manipulated by the various gates in the LSTM node in order to retain important information or lose unimportant data. The *forget gate* is used to discard information from the *cell* vector, whereas the *input gate* adds new information to the *cell*. The *output gate* determines the aspect of the new information, passed into the RNN and modified by the *cell*, to output as h_t. The vectors can also be expressed formally as follows

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{5}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{6}$$

$$C_t = f_t * C_{t-1} + i_t * tanh(W_C[h_{t-1}, x_t] + b_C) \tag{7}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t * tanh(C_t) \tag{9}$$

with W as weight variables, b as biases, and σ as the sigmoid function. Figure 4 shows a depiction of the LSTM cell.

As a result of their size and complexity, RNNs are particularly hard to train and can become subject to vanishing gradients. Vanishing gradients occur when states in later layers of a neural network are not influenced by the changes on previous layers. Vanishing gradients prevent the network, as a whole, from learning long-term time dependencies [23] such as a compliant response observed early in a long video sequence. One solution for this problem is batch normalization, a regularization method that uses the mean and variance of mini-batch updates to normalize layer inputs in CNNs [23]. A recent study has expanded this idea to generate a batch normalized recurrent neural network by normalizing the inputs into an LSTM cell [24]. The batch normalized LSTM (BN-LSTM) cell allows normalization in large RNNs. This allows changes to the current time step to occur based on information potentially thousands of time steps in the past. We use the implementation described in [24] to normalize the inputs for our LSTM cells.

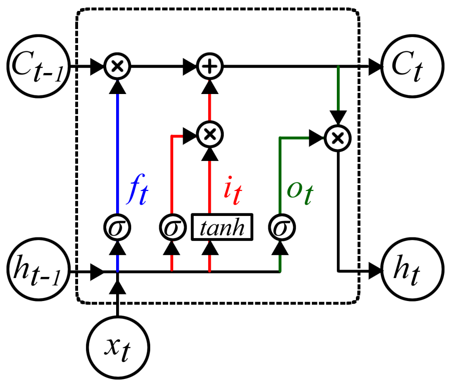

Fig. 4. An LSTM cell. The forget gate is indicated in blue, the input gate in red, and the output gate in green. The sigmoid function is indicated with σ and multiplication and addition functions are represented using $\times$ and $+$ respectively.

III. TRAINING

In order to train our DRQN we performed an IRB approved user study with a tele-operated robot. The DRQN was implemented using Tensorflow and we used ROS to control the robot. Our system can be located at [25].

A. Data Collection

We collected demonstration data to train our DRQN through a user study in which a tele-operated NAO humanoid robot interacted with users. The user study consisted of 11 students from the University of New Hampshire (10 male and 1 female). The participants performed a total of 20 sessions with the tele-operated robot as it delivered a social greeting BI. Participants were asked to provide an equal number of compliant and non-compliant responses. In order to reduce the risk of participant fatigue and in order to generate significant data we restricted the length of the sessions so that no more than two prompts could be delivered. A noncompliant case was one in which the participant failed to respond to either prompt and a compliant response was any session in which either the first or second prompt was positively responded to. During the user study, all participants were aware that the robot was tele-operated. Observation data was collected using the NAO's camera, the NAO's microphone, and a depth camera situated to the left of the robot (Figure 1).

After curation of our dataset to remove examples that were erroneous, ambiguous, or included video that was shorter than 10 frames in length, we possessed 105 compliant and 102 non-compliant sessions. We preprocessed the data in order to highlight regions of interest and reduce noise. Video taken using the NAO's camera was cropped to foveate the view of the participant's face and head (Figure 2(c)). When a face could not be identified the image was left unaltered. We also processed the audio input by passing the dry audio data through a finite impulse response (FIR) filter to reduce noise. FIR filters smooth and reduce the noise of one dimensional signal data. The filtered data was then converted into a spectrogram (Figure 2(e)) which was separated into a number of frames equal in length to the video and depth data. Spectrograms are two dimensional representations of audio frequencies over time [26]. Spectrograms have seen significant use in many deep recurrent network architectures that process or generate sound data. Finally we doubled the number of input demonstrations by duplicating each input file with horizontally mirrored video and depth data. The filtered audio spectra were not altered in the mirrored examples because spectrograms are defined by their orientation.

B. *DRQN Structure*

Our DRQN processes observation data as a collection of frames. A single frame consists of a 3-channel image taken from the NAO's camera, a single-channel point cloud array from the depth camera (Figure 2(d)), and a 3-channel spectrogram of the audio data. In order to reduce the number of parameters in our system, each frame was re-sized to be 80 pixels in both height and width prior to being read into the DRQN. We also pass an additional parameter to our model in order to define the state: the number of prompts that had been given so far in the interaction. Calling the *abort* action is dependent on the length of the interaction, a factor that cannot be discerned from the raw pixel data but is still integral to our state definition.

Our DRQN network architecture can be observed in Figure 3. The structure was modeled after the network described in [19]. We use three CNN stacks, one for each type of data being processed. The stacks are identical in structure but are composed of independent weights and biases. Each stack is composed of three convolutional layers separated by ReLU activation layers. After the data from the frame has passed through the CNN stacks, the outputs are merged together and passed to two stacked BN-LSTM layers. The BN-LSTM layers identify time relevant data that occurs across many frames. The output of the final BN-LSTM is combined with the number of prompts that have been delivered in the BI so far. These combined values are passed into a fully connected layer that outputs the Q-values for each action. Once the entire response has been parsed, we execute the action that has the highest Q-value. We include the number of prompts at the end of our network since the value is neither multi-dimensional nor does its expression vary over time.

Because our system required live subjects to be trained, we were unable to take the typical reinforcement learning approach of repeatedly sampling experience data from the test problem. We instead configured the demonstrations we obtained in our user study to resemble experience replay. Because of the small datasets that DQNs are exposed to, they are typically trained with an adaptive learning rate optimizer. We selected the ADAM [27] optimizer as it builds upon other adaptive learning rate optimizers such as RMSProp and ADAdelta [28].

IV. EVALUATION

We evaluated our system by using leave one out cross-validation to assess the models generalizability. Our model was trained using mini-batches of 10 demonstrations delivered over the course of 38 epochs. To perform the cross-validation, we removed all of the examples of a specific participant from the training dataset and then evaluated them after the model had finished optimizing. Figure 5 shows the accuracies obtained by the DRQN model when evaluating each participant. We obtained an average accuracy of 62.8% from our cross-validation. Based on our demonstration data we found that a random policy will be accurate 43.3% of the time, therefore, our model shows an increase of approximately 150% in its ability to correctly generalize an action from unknown data.

The highest value we observed in our cross-validation was 83.3%. This indicates that the examples generated with participant 7 coincided with the features our model associated with compliant and non-compliant interactions. It also shows that should our model be exposed to the entire training dataset it is likely to obtain a very high level of accuracy. Conversely, the accuracy obtained from the model tested with participant 8's examples was the lowest (43.2%), and was comparable to selecting actions randomly. The result from participant 8's model was interesting in that the training dataset had yet to converge by the 38th epoch, unlike models that were trained with participant 8's examples. This implies that the examples generated with participant 8 provided several strong features that were easily learnt by other models. When those examples were removed from the training set, it was difficult for a robust model to be generated. However, we believe that with further training epochs important features could have been identified by the optimizer, increasing the model's accuracy.

In our study we found a trend that indicated a larger number of filters in the convolutional layers improved the overall generalizability of the system. The improved accuracy likely results from a greater expression of the various features in the demonstration data. It is very likely that increasing the amount of training data would improve the accuracy of the system, but we consider our model's accuracy, given our very limited set of examples (further limited when performing cross-validation), to be promising. Modifications that vary the data could also generate a more robust trained model (ie. brightness, scaling, different audio filters).

V. CONCLUSIONS

Feature based robotics applications suffer from several drawbacks including reduced generalizability and the potential for information to be lost as a result of designer bias. Deep Q-learning provides an alternative to the challenges of perception in that it can identify significant features in large image datasets and is able to generate a policy from a few examples. As a result, deep Q-learning seems an appropriate

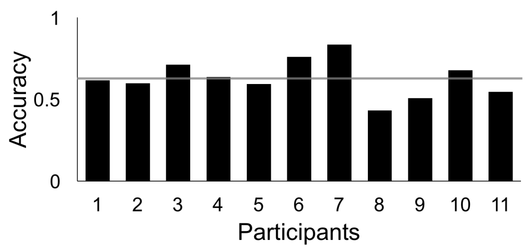

Fig. 5. Cross-validation results of our DRQN. The accuracy of each model when trained without the indicated participants is shown in black. The grey line represents the average accuracy of the cross-validation.

tool to solve many LfD problems. We reinforced this claim by using a DRQN model to generalize the features important in a robot mediated BI with greater than 80% accuracy. Our future goal is to now deliver an automated version of our behavioral therapy using the generated DRQN model. We also hope to further improve our model's accuracy through the use of transfer learning. Transfer learning allows neural networks to use the feature information learned by networks trained on large data sets such as ImageNet [29]. We will also investigate the use of both optical flow and visual attention towards improving our models predictive capability.

Having indicated the potential for deep Q-learning to solve robotics problems, we also intend to investigate its effectiveness on other, more complex LfD domains. We would like to improve our existing system further by removing the restrictive assumptions we place on our problem (e.g. limits on the number of prompts performed). By investigating the roles that time constraints place on certain action executions, we can clearly define when actions should and should not be performed.

ACKNOWLEDGMENT

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

REFERENCES

[1] J. Alper and S. Olson, "Report to the president realizing the full potential of health information technology to improve healthcare for americans: The path forward," *Healthcare Information and Management Systems Society*, 2010.

[2] P. Pennisi, A. Tonacci, G. Tartarisco, L. Billeci, L. Ruta, S. Gangemi, and G. Pioggia, "Autism and social robotics: A systematic review," *Autism Research*, 2015.

[3] J.-J. Cabibihan, H. Javed, M. Ang, and S. M. Aljunied, "Why robots? a survey on the roles and benefits of social robots in the therapy of children with autism," *International journal of social robotics*, vol. 5, no. 4, pp. 593–618, 2013.

[4] J. Wainer, B. Robins, F. Amirabdollahian, and K. Dautenhahn, "Using the humanoid robot kaspar to autonomously play triadic games and facilitate collaborative play among children with autism," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 3, pp. 183–199, 2014.

[5] T. Esubalew, U. Lahiri, A. R. Swanson, J. A. Crittendon, Z. E. Warren, N. Sarkar *et al.*, "A step towards developing adaptive robot-mediated intervention architecture (aria) for children with autism," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 2, pp. 289–299, 2013.

[6] S. Chernova and A. L. Thomaz, "Robot learning from human teachers," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.

[7] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[8] K. Bullard, B. Akgun, S. Chernova, and A. L. Thomaz, "Grounding action parameters from demonstration," in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*. IEEE, 2016, pp. 253–260.

[9] C. Chao, M. Cakmak, and A. L. Thomaz, "Towards grounding concepts for transfer in goal learning from demonstration," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011, pp. 1–6.

[10] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, "Learning grounded finite-state representations from unstructured demonstrations," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015.

[11] R. Cubek, W. Ertel, and G. Palm, "High-level learning from demonstration with conceptual spaces and subspace clustering," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2592–2597.

[12] A. Mohseni-Kabir, C. Rich, S. Chernova, C. L. Sidner, and D. Miller, "Interactive hierarchical task learning from a single demonstration," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2015, pp. 205–212.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] K. Young, R. Hayward, and G. Vasan, "NeuroHex: A deep Q-learning hex agent," *arXiv preprint arXiv:1604.07097*, 2016.

[15] V. Firoiu, W. F. Whitney, and J. B. Tenenbaum, "Beating the world's best at super smash bros. with deep reinforcement learning," *arXiv preprint arXiv:1702.06230*, 2017.

[16] K. Narasimhan, T. Kulkarni, and R. Barzilay, "Language understanding for text-based games using deep reinforcement learning," *arXiv preprint arXiv:1506.08941*, 2015.

[17] R. M. Foxx, "Applied behavior analysis treatment of autism: The state of the art," *Child and adolescent psychiatric clinics of North America*, vol. 17, no. 4, pp. 821–834, 2008.

[18] M. Begum, R. W. Serna, D. Kontak, J. Allspaw, J. Kuczynski, H. A. Yanco, and J. Suarez, "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, 2015, pp. 335–342.

[19] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," *arXiv preprint arXiv:1507.06527*, 2015.

[20] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling." in *Interspeech*, 2012, pp. 194–197.

[21] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[22] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[24] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," *arXiv preprint arXiv:1603.09025*, 2016.

[25] M. Clark-Turner, "DQN-ASD," 2017. [Online]. Available: https://github.com/AssistiveRoboticsUNH/DQN-ASD

[26] H. Harutyunyan and H. Khachatrian, "Combining cnn and rnn for spoken language identification," 2016. [Online]. Available: https://yerevann.github.io/2016/06/26/combining-cnn-and-rnn-for-spoken-language-identification/

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] S. Ruder, "An overview of gradient descent optimization algorithms," 2016. [Online]. Available: http://sebastianruder.com/optimizing-gradient-descent/index.html#gradientdescentvariants

[29] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.