INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Speech Driven Backchannel Generation using Deep Q-Network for Enhancing Engagement in Human-Robot Interaction

Nusrah Hussain, Engin Erzin, T. Metin Sezgin, Yücel Yemez

Koç University, Turkey
{nhussain15,eerzin,mtsezgin,yyemez}@ku.edu.tr

Abstract

We present a novel method for training a social robot to generate backchannels during human-robot interaction. We address the problem within an off-policy reinforcement learning framework, and show how a robot may learn to produce non-verbal backchannels like laughs, when trained to maximize the engagement and attention of the user. A major contribution of this work is the formulation of the problem as a Markov decision process (MDP) with states defined by the speech activity of the user and rewards generated by quantified engagement levels. The problem that we address falls into the class of applications where unlimited interaction with the environment is not possible (our environment being a human) because it may be time-consuming, costly, impracticable or even dangerous in case a bad policy is executed. Therefore, we introduce deep Q-network (DQN) in a batch reinforcement learning framework, where an optimal policy is learned from a batch data collected using a more controlled policy. We suggest the use of human-to-human dyadic interaction datasets as a batch of trajectories to train an agent for engaging interactions. Our experiments demonstrate the potential of our method to train a robot for engaging behaviors in an offline manner.

Index Terms: human-robot interaction, engagement, backchannels, reinforcement learning

1. Introduction

Interaction of a robot with a human in a domestic environment needs to be characterized by behaviors and norms compatible with humans for a successful interaction experience. In applications like companionship, tutoring, and ambient assisting living, actions which encourage attention and engagement of the user are necessary for an effective interaction. A survey by Clavel et. al. summarizes the issues regarding engagement in human-agent interactions, emphasizing its importance and indicating the growing interest of researchers in the field [1]. Backchannels like non-verbal gestures (nods and smiles), non-verbal vocalizations (mm, uh-huh, laughs) and verbal expressions (yes, right) are an important aspect of engagement and have been shown to promote engagement and interest levels of the user [2, 3]. Researchers have mainly focused on rule-based backchannel generation [4, 5] or data-driven unsupervised methods [6]. In this work, we show how to formulate the problem in a reinforcement learning framework and train an agent to learn a policy for backchannel generation that maximizes the engagement of the user. To the best of our knowledge, this problem has not been addressed before in the context of reinforcement learning.

Reinforcement learning (RL) has shown much success in the past few years as an optimization algorithm for problems having temporal structure and seems a natural choice for a variety of problems in robotics. Several works exist, which have used reinforcement learning to impart human-like behaviors into social robots. Qureshi et. al. [7] [8] use multi-modal DQN to train a robot to greet like humans with the sequential actions of wait, look, wave and shake hand. The reward comes from successful handshakes via a sensor. RL is employed to adjust motion speed, timing, interaction distances, and gaze in the context of human-robot interaction (HRI) by Mitsunaga et. al. [9]. The reward is based on the amount of movement of the subject and the time spent gazing at the robot in one interaction. Lathuilière et. al. [10] use recurrent neural network architecture in combination with Q-learning to find an optimal policy for robot gaze control in HRI. In these works, however, the agent either interacts with the environment (humans) for several days or training is done using simulators. In our work, we address the challenge where experience on a real physical system may be tedious to obtain, expensive, time-consuming and hard to simulate. We propose to use human-to-human interaction datasets as a batch of off-policy samples (trajectories) and use them in the context of offline batch reinforcement learning. The goal is to learn from the batch data the sequence of actions (or trajectories) that result in higher engagement and to formulate a policy around those regions of state-action space. Since we do not aim to mimic the dataset, instead of typical supervised learning metrics (like accuracy, recall, precision etc.) we evaluate our training using Bellman residual and expected return from the new policies.

The contribution of this work is two fold:

1. We propose a reinforcement learning formulation for backchannel generation that enhances engagement levels of the user during human-robot interaction. States are generated using speech features of the user and rewards are calculated by the engagement levels of the user. We demonstrate the experiments on laughs as a non-verbal backchannel.
2. We present the use of pre-recorded human-to-human dyadic interaction datasets as a batch of samples acquired by a behavior policy (another human). The optimal Q-value function is extracted from this batch data using a modified version of deep Q network (DQN) as a model-free value-based batch-RL algorithm. A greedy policy can implicitly be deduced from the optimal Q-value function.

2. Related Work

2.1. Engagement in Interactions

Several definitions of engagement exist in the literature, which have been described in detail by Glas et. al. [11]. Poggi describes engagement as: “the value that a participant in an interaction attributes to the goal of being together with the other participant(s) and of continuing the interaction” [12]. One of the pioneering studies on measurement of engagement is the work by Rich et. al. [13], where the authors propose an engagement model for collaborative interactions between human and computer. They define four types of events as engagement in-

http://dx.doi.org/10.21437/Interspeech.2019-2521

dicators, referred to as connection events (CEs), which include directed gaze, mutual facial gaze, adjacency pair, and backchannels. Directed gaze event is defined when both participants look at a nearby object related to the interaction at the same time. The mutual facial gaze occurs when there is face-to-face eye contact. Adjacency pair indicates a successful event when turn taking occurs with some minimal time gap. Finally, backchannels refer to the generation of audio-visual feedback by a listener during the speaker's turn. In our work, we use these connection events to quantify engagement and generate a single scalar value at each time step to represent the rewards. An alternative option may be to directly annotate the engagement levels in the dataset. However, automatic detection of engagement allows the refinement of the policy in the future by continuously updating the policy as the agent interacts with humans.

2.2. Reinforcement Learning

In general, the reinforcement learning formulates the optimization problem as a Markov decision process (MDP) (S, A, p, r, γ) in which the environment has a state $s \in S$, the agent takes an action $a \in A$ and the scalar reward $r(s, a) \in \mathbb{R}$ is generated by the environment. The transition dynamics $p(s_{t+1}|s_t, a_t)$ gives the probability of next state s_{t+1} given that at time t the state s_t is observed and action a_t is taken. The discount factor $\gamma \in [0, 1)$ weighs the future rewards, determining the extent of temporal data that is affected by the current action. The solution to any MDP is a policy $\pi(a|s)$ which maximizes the expectation of sum of discounted rewards, i.e., the return.

Reinforcement learning is more commonly found as an online learning algorithm where the agent updates its policy while it interacts with the environment. However, RL may also be formulated as a batch (offline) or semi-batch learning technique, as described at length in a survey paper [14]. Batch reinforcement learning refers to the reinforcement learning setting where the task is to learn an optimal policy from a fixed batch of transitions sampled with some behavioral policy [15]. Our interest in batch reinforcement learning is mainly due to the difficulty of human-robot interaction over extensive time lengths and under varying conditions. Moreover, several recordings of human-to-human dyadic interaction datasets are readily available. Thus, we utilize these recordings as a batch of samples collected by another policy and extract the optimum policy for our goal.

Neural fitted Q-iterations (NFQ) is a well-known batch reinforcement learning algorithm [16] based on Q-learning. It trains a neural network by fitting the Bellman optimality equation [17] given by

$$Q^*(s, a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} Q^*(S_{t+1}, a')|S_t = s, A_t = a]. \tag{1}$$

A more recent variant of the online Q-learning is the deep Q-network [18] which introduces two further concepts within the Q-learning approach. First, it uses experience replay to randomize over data in order to de-correlate sequential steps and secondly the target values, towards which the Q-values are iteratively updated, are refreshed periodically. Some works like [19] use DQN with demonstrations to accelerate the learning process. Similarly, we introduce the batch version of DQN (batch-DQN) and show its superiority over NFQ. We describe our batch-DQN technique in more detail in Section 3.6.

3. Proposed Method

3.1. Dataset

We work with the IEMOCAP dataset [20], which is designed to analyze expressive human interactions. It consists of five sessions acted by ten professional actors performing dyadic human-to-human conversations. In total, there are 151 dialogs on 8 hypothetical scenes and 3 scripted plays, performed in pairs of the opposite gender. This dataset represents human behavior in a variety of situations where the policy behind each dialog may vary.

We define our problem by assuming that, of the two actors, one represents the behavior policy (i.e., generates actions in the form of backchannels) while the second actor plays the role of the environment generating states and rewards. Considering the IEMOCAP dataset as a batch of trajectories collected by a behavioral policy, we apply our batch reinforcement learning method to extract an optimal policy that maximizes engagement during human-robot interaction.

3.2. Markov Decision Process

Though our framework is general for any type of event generation, we have conducted experiments to see the effectiveness of batch reinforcement learning on laughter generation. Given a state of the environment, the agent needs to make a decision on whether a backchannel event at that instance will contribute to engagement. We define the states, actions, and rewards as follows:

- **State:** The state of the environment is represented by speech features extracted from past one second of data at every 25 msec step. This produces state information at a rate of 40 Hz. The dimension of the extracted feature is 209, which is described in detail in Section 3.4.
- **Action:** Agent's action is a binary variable, indicating the absence or presence of the backchannel. The backchannels of the user with behavioral policy in the dataset were labeled at a rate of 40 Hz.
- **Reward:** The reward is a scalar quantity which comes from the engagement measures of the user at every time step. The quantification of engagement from the dataset via Sidner's method is detailed in Section 3.5.

3.3. Generation of Tuples

Batch reinforcement learning algorithms work with tuples of the form $\langle s_t, a_t, r_t, s_{t+1}\rangle$ for $t = 1 : T$. At time t, s_t is the state of the environment, a_t is the action of laughing by the agent and r_t is the reward defined in terms of engagement levels generated from the environment. Figure 1 shows the time windows used to extract states, rewards, and actions in one tuple. The dataset is pre-processed and such tuples are saved in a buffer.

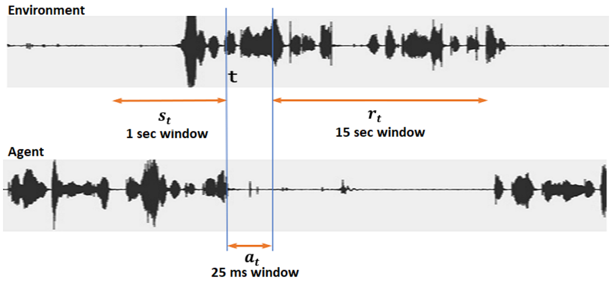

Figure 1: *Reinforcement learning formulation of speech driven backchannel generation (not drawn to scale)*

3.4. Speech Features

The states are defined using the mel-frequency cepstrum coefficients (MFCCs) and prosody features extracted from the speech signal of the environment. 13-dimensional MFCC features are computed using 40 milliseconds sliding Hamming window at intervals of 25 milliseconds. The speech intensity, pitch, and confidence-to-pitch with their first derivates make up a 6-dimensional prosody feature, so a 19-dimensional feature vector is formed when MFCCs and prosody are concatenated as in [21]. Following this, feature summarization is performed where a set of statistical quantities are computed that describe the short-term distribution of each feature over the past one second. These quantities comprise eleven functions, more specifically mean, standard deviation, skewness, kurtosis, range, minimum, maximum, first quantile, third quantile, median quantile and inter-quartile range, which were successfully used before by [22]. The dimension of each of these statistical feature vectors is 11 times the dimension of the corresponding feature vector. This makes the feature size of length 209.

3.5. Engagement Measurement

Our measure of engagement is based on the method proposed in [13], which is applicable to face-to-face collaborative HCI scenarios. Similar to the description in Section 2.1, we use the connection events (CE) (1) mutual facial gaze, (2) adjacency pair and (3) backchannels (that include laughs, smiles, nods and head-shakes) to quantify engagement. In [13], the 'directed gaze' event is defined when the agent and the participant look at a nearby object related to the interaction at the same time. However, in our dataset, since we do not have objects of interest at which both parties look at, we exclude it in our definition. The extracted CEs are then used to calculate a summarizing engagement metric called 'mean time between connection events' (MTBCE). MTBCE measures the frequency of successful connection events that is for a given time interval T, MTBCE is calculated by T / (no. of CEs in T). As MTBCE is inversely proportional to engagement, similarly to [13], we use pace = 1/MTBCE to quantify the engagement between a participant and the robot. The pace measure is calculated over a window of 15 seconds in our experiments. Since human engagement varies slowly over time, window size of 15 seconds was chosen after some analysis to avoid abrupt changes in rewards.

3.6. Batch-DQN

While the existing batch RL techniques have presented their success in a number of settings, they do not efficiently scale to large datasets. When dealing with large amounts of patterns, the question arises whether all patterns need to be used in every training step and whether there exists a way in which only parts of the patterns can be selected while still allowing for successful training and good policies [23]. Contrary to existing batch-RL techniques, we initialize an experience replay buffer much smaller than the batch data. During training, mini-batches of data are taken and only the samples that agree with current ϵ-greedy policy are pushed into the buffer, where ϵ is the probability of choosing the action present in the batch and $(1-\epsilon)$ is the probability of choosing an action greedily. As ϵ decays with time, the batch converges to those samples that are more likely to be seen when following the optimal policy. Like DQN, at every iteration, a random mini-batch is sampled from the replay buffer and updates are performed using the Bellman control equation.

4. Experimental Setup

From approximately 10 hours of recordings in IEMOCAP, $\langle s_t, a_t, r_t, s_{t+1}\rangle$ tuples form a batch data of size 1.5 million (as described in Section 1). We further double the batch data by switching the roles of environment and behavior policy, hence increasing the batch size to approximately 3 million tuples. We define the train and test sets in the ratio 4:1 as leave one subject out (LOSO), hence 5 folds of training are performed which are subject independent. Training is done for the two techniques: batch-DQN and neural fitted Q-iterations (NFQ).

We model the Q-function approximation network with a multi-layer perceptron of two hidden layers and use ReLU as the nonlinear activation function. The Q-network receives an input feature length 209, followed by hidden layers of sizes 100 and 25, and produces two Q-values so as to generate a backchannel or to remain inactive. The smooth-L1 loss from the Bellman control equation is minimized using Adam optimizer. We have used a discount factor value 0.99 in all experiments.

4.1. Policy Evaluation Metrics

Evaluation of the resultant policy is a challenging problem since the environment (i.e., the human participant) is not readily available in our case. Although it is possible to conduct experiments with human-robot interactions, it is desirable to first understand the policy's effectiveness using quantitative measures. Before presenting the results, we first describe below the metrics used for evaluation.

4.1.1. Bellman Residual

For a Q-value function approximation network Q_θ, the Bellman residual is defined as the difference between the two sides of a Bellman control equation [24]. A smaller residual error means that the learned policy is closer to the optimal policy and is a true Q-function since it follows the Bellman equation more closely. Similar to the work of [15], we compute the Bellman residual, B_r, over the entire batch of data as

$$B_r = \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}}(Q_\theta(s_t,a_t) - [r_t+\gamma * \max_{a\in A} Q_\theta(s_{t+1},a)])^2. \quad (2)$$

4.1.2. Off-Policy Policy Evaluation (OPE)

Off-policy policy evaluation (OPE) is defined as the problem of estimating the value V^π of a new policy using the batch of samples collected independently from a behavior policy. In the last few years, many OPE techniques have emerged because of its importance in cases where a new policy cannot be tested directly with the environment [25, 26]. To compare the values of policies resulting from NFQ and batch-DQN, we applied the step-wise weighted importance sampling estimator (WIS) given by

$$\hat{V}^{\pi}_{step-WIS} = \sum_{i=1}^{n}\sum_{t=0}^{T-1}\gamma^t \frac{\rho_t^{(i)}}{\sum_{i=1}^{n}\rho_t^{(i)}} r_t^{(i)}, \quad (3)$$

where n is the number of trajectories, T is the length of each trajectory and γ is the discount factor. Then, the importance weight ρ is defined as the ratio of the probability of the first $t+1$ steps of a trajectory under π to the probability under a behavior policy π_b and is given as $\rho_t = \prod_{i=0}^{t}\frac{\pi(a_i|s_i)}{\pi_b(a_i|s_i)}$. The importance sampling approach to evaluation relies on using the importance weights ρ_t to adjust for the difference between the probability of a trajectory under the behaviour policy π_b and the probability under the evaluation policy π. To perform this evaluation we defined trajectories as frames of length $T = 100$

with shifts of one sample. Following discussion of the work in [27], the behavior policy π_b was estimated using approximate nearest neighbor [28].

4.1.3. Naturalness of laughs

Another metric we take into consideration is similarity of laughs to that of a human. For example, a policy that results in laughs which lasts several minutes would not be natural and would result in discomfort of the user. Similarity can be analyzed by other characteristics like how a laugh is generated and how it sounds but that is beyond the scope of this paper. We only focus on the lengths of laugh here. In order to assess the naturalness of the laughs generated for the agent, we analyze the distribution of laughter duration present in the dataset. A successful agent should be able to produce similar statistics. To measure the similarity between two probability histograms we used symmetric Kullback-Leibler Divergence measure as well as statistical metrics like mean, max and inter-quartile range.

5. Results & Discussion

To illustrate the effectiveness of batch-DQN over NFQ, Figure 2 shows the Bellman residuals plotted against the epoch number for one fold of the laughter training. We observe that the error reduces when a separate target network is introduced, and improves further when samples are randomly selected from the replay buffer which contains tuples complying with the current epsilon-greedy policy. A closer look at the minimum Bellman residuals when averaged over all folds of the training shows the superiority of batch-DQN. Table 1 gives the mean errors obtained over the test sets.

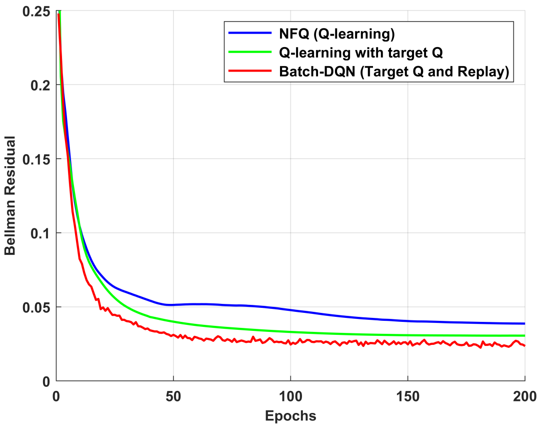

Figure 2: *Bellman residual vs training epochs*

Table 1: *Bellman residuals for test set over 5-fold training*

Technique	Bellman Residual
Neural Fitted Q-Iterations	0.0571 ± 0.0097
Batch-DQN	0.0371 ± 0.0130

The off-policy policy evaluation results obtained from stepwise weighted importance sampling are shown in Table 2. The behavior policy is modelled with approximate nearest neighbor algorithm while the policies learned from NFQ and batch-DQN are deduced from the corresponding Q-networks such that 0.9 probability is assigned to the action (laughter or non-laughter) suggested by the greedy policy. The values returned by OPE represent the expected cumulative discounted rewards estimated for each method. Ideally these values need to be computed over infinite lengths but keeping in mind the numerical limitations we calculated them over trajectories of length 250 samples. While both techniques perform better than the behavior policy baseline, it can be noted that the policy obtained by our batch-DQN method outperforms the policy learned by the NFQ technique.

Table 2: *Off-policy policy evaluation (OPE) results*

Technique	Estimated V^π
Dataset (Behavior Policy)	23.15
Neural Fitted Q-Iterations	24.32
Batch-DQN	**27.57**

Finally, Table 3 shows the statistical similarity of laughter duration generated by each policy to that of a human. While the mean is comparable to a human, the prominent weakness of NFQ policy is seen by the maximum length of the laugh it generates. NFQ results in laughs many folds longer than a typical human laugh ($\sim$ 198 sec). A closer look at the reward distribution in the dataset reveals that the mean reward is higher in the intervals where laughs are present. This explains the reason why both techniques prefer to generate laughs more frequently than in the dataset. This can be improved upon in the future work by imposing some form of constraints to make the lengths of laughs more natural. The KL-divergence value also shows the superiority of batch-DQN over NFQ.

Table 3: *Similarity of agent laughs to human laughs*

	Laugh Duration (sec)			KL-Divergence
	Mean	Max	Std	
Human	0.95	9.17	0.89	-
NFQ	1.15	198.45	3.18	0.2890
Batch-DQN	0.76	46.45	0.99	0.1921

6. Conclusion

We have presented a scheme to train a robot for backchannel generation in a human-robot interaction, so as to maximize the engagement of the user. The formulation of this problem as a reinforcement learning problem allows the robot to learn the effect of an action in current time step on the entire future interaction. We have also shown how the available datasets on human-to-human interaction may be used as a batch of off-policy trajectories. Our experiments have shown the advantage of our batch-DQN algorithm over neural fitted Q-iterations technique which can be considered as a baseline for batch RL. An immediate extension of this work is to perform subjective evaluations with human participants. Furthermore, this work may be extended by enriching the state definition using visual features and by training for other forms of backchannels.

7. Acknowledgements

This work is funded by Turkish Scientific and Technical Research Council (TUBITAK) under grant number 217E040 and the first author is supported by Higher Education Commission (HEC) Pakistan.

8. References

[1] C. Clavel, A. Cafaro, S. Campano, and C. Pelachaud, "Fostering user engagement in face-to-face human-agent interactions: a survey," in *Toward Robotic Socially Believable Behaving Systems-Volume II*. Springer, 2016, pp. 93–120.

[2] B. B. Türker, Z. Buçinca, E. Erzin, Y. Yemez, and M. Sezgin, "Analysis of engagement and user experience with a laughter responsive social robot," in *Proc. 18th Annu. Conf. Int. Speech Commun. Assoc*, 2017, pp. 844–848.

[3] B. Inden, Z. Malisz, P. Wagner, and I. Wachsmuth, "Timing and entrainment of multimodal backchanneling behavior for an embodied conversational agent," in *Proceedings of the 15th ACM on International conference on multimodal interaction*. ACM, 2013, pp. 181–188.

[4] S. Al Moubayed, M. Baklouti, M. Chetouani, T. Dutoit, A. Mahdhaoui, J.-C. Martin, S. Ondas, C. Pelachaud, J. Urbain, and M. Yilmaz, "Generating robot/agent backchannels during a storytelling experiment," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3749–3754.

[5] C. Liu, C. T. Ishi, H. Ishiguro, and N. Hagita, "Generation of nodding, head tilting and eye gazing for human-robot dialogue interaction," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2012, pp. 285–292.

[6] H. Admoni and B. Scassellati, "Data-driven model of nonverbal behavior for socially assistive human-robot interactions," in *Proceedings of the 16th international conference on multimodal interaction*. ACM, 2014, pp. 196–199.

[7] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Robot gains social intelligence through multimodal deep reinforcement learning," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 745–751.

[8] ——, "Show, attend and interact: Perceivable human-robot social interaction through neural attention q-network," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1639–1645.

[9] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning," *Journal of the Robotics Society of Japan*, vol. 24, no. 7, pp. 820–829, 2006.

[10] S. Lathuilière, B. Massé, P. Mesejo, and R. Horaud, "Neural network based reinforcement learning for audio–visual gaze control in human–robot interaction," *Pattern Recognition Letters*, vol. 118, pp. 61–71, 2019.

[11] N. Glas and C. Pelachaud, "Definitions of engagement in human-agent interaction," in *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2015, pp. 944–949.

[12] I. Poggi, *Mind, hands, face and body: a goal and belief view of multimodal communication*. Weidler, 2007.

[13] C. Rich, B. Ponsler, A. Holroyd, and C. L. Sidner, "Recognizing engagement in human-robot interaction," in *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*. IEEE, 2010, pp. 375–382.

[14] S. Lange, T. Gabel, and M. Riedmiller, "Batch reinforcement learning," in *Reinforcement learning*. Springer, 2012, pp. 45–73.

[15] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 503–556, 2005.

[16] M. Riedmiller, "Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method," in *European Conference on Machine Learning*. Springer, 2005, pp. 317–328.

[17] R. S. Sutton and A. G. Barto, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[18] 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, p. 529, 2015.

[19] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband *et al.*, "Deep q-learning from demonstrations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, p. 335, 2008.

[21] E. Bozkurt, E. Erzin, and Y. Yemez, "Multimodal analysis of speech and arm motion for prosody-driven synthesis of beat gestures," *Speech Communication*, vol. 85, pp. 29–42, December 2016.

[22] A. Metallinou, A. Katsamanis, and S. Narayanan, "Tracking continuous emotional trends of participants during affective dyadic interactions using body language and speech information," *Image and Vision Computing*, vol. 31, no. 2, pp. 137–152, 2013.

[23] M. Plutowski and H. White, "Selecting concise training sets from clean data," *IEEE Transactions on neural networks*, vol. 4, no. 2, pp. 305–318, 1993.

[24] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 30–37.

[25] P. Thomas and E. Brunskill, "Data-efficient off-policy policy evaluation for reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 2139–2148.

[26] S. Doroudi, P. S. Thomas, and E. Brunskill, "Importance sampling for fair policy selection." *Grantee Submission*, 2017.

[27] A. Raghu, O. Gottesman, Y. Liu, M. Komorowski, A. Faisal, F. Doshi-Velez, and E. Brunskill, "Behaviour policy estimation in off-policy policy evaluation: Calibration matters," *arXiv preprint arXiv:1807.01066*, 2018.

[28] V. Hyvönen, T. Pitkänen, S. Tasoulis, E. Jääsaari, R. Tuomainen, L. Wang, J. Corander, and T. Roos, "Fast nearest neighbor search through sparse random projections and voting," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 881–888.