

Engaging Human-Robot Interaction with Batch Reinforcement Learning

by

Nusrah Hussain

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of
Doctor of Philosophy

in

Electrical and Electronics Engineering



KOÇ ÜNİVERSİTESİ

September 08, 2020

Engaging Human-Robot Interaction with Batch Reinforcement Learning

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Nusrah Hussain

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Engin Erzin (Advisor)

Prof. Yücel Yemez (Co-Advisor)

Assoc. Prof. Metin Sezgin

Prof. Murat Tekalp

Prof. Ali Albert Salah

Assist. Prof. Anca Dragan

Date: _____



Dedicated to my husband.

ABSTRACT

Engaging Human-Robot Interaction with Batch Reinforcement Learning

Nusrah Hussain

Doctor of Philosophy in Electrical and Electronics Engineering

September 08, 2020

A common issue in the field of social robotics is the need to maintain user engagement during human-robot interaction (HRI). Engagement has been used as a typical metric to gauge the success of HRI, and hence it is regarded as a universal goal in the design of social robots. In this thesis, we train a generation model of non-verbal behaviors, smiles and nods, as backchannels in a robot to engage humans during HRI. We propose a novel batch reinforcement learning (batch-RL) formulation for the task, where we take advantage of recorded human-to-human interaction data to learn a policy offline. The formulation treats user engagement as the reward and constructs a backchannel policy that maximizes it. We propose three value-based off-policy batch-RL algorithms to address the problem, which differ in the manipulation of the samples in the dataset to make the gradient updates. To evaluate the policies trained with these algorithms, offline evaluation methods are used such as off-policy policy evaluation (OPE) and Bellman residual. A final work presented in the thesis is the design and execution of a user study on HRI with a backchanneling robot. The interaction is designed with an expressive 3d robotic head in a story-shaping interaction scenario, where the learned backchannel policy controls the nod and smile behaviors. Subjective questionnaires and engagement values extracted from user's social signals are used to assess the impact of robot's social behavior on the participants. The higher acceptability of the RL policy versus a baseline policy is indicated by the statistically significant differences in the evaluation scores. The research work presented in this thesis addresses only one class of robot behavior towards socially engaging robots. As a pioneering work, it paves way for automation of numerous other desired robot behaviors that target other metrics used in the design of human-robot interaction systems.

ÖZETÇE

KAYITLI VERİDEN PEKİŞTİRMELİ ÖĞRENME YOLUYLA İNSAN ROBOT ETKİLEŞİMİNDE KULLANICI İLGİSİNİN ENİYİLENMESİ

Nusrah Hussain

Elektrik ve Elektronik Mühendisliği, Doktora

Eylül 08, 2020

Sosyal robot bilimi alanında yaygın bir sorun, insan-robot etkileşimi (İRE) sırasında kullanıcı ilgisinin sürdürülebilmesidir. İlgı seviyesi etkileşimin başarısını ölçmek için sıklıkla kullanılan bir ölçüttür, ve sosyal robotların tasarımında evrensel bir amaç olarak kabul edilir. Bu tez çalışmasında, İRE için bir robota ait sözel olmayan gülümseme ve kafa sallama jestlerini içeren arka kanal davranışlarını üreten bir model eğitmeyi hedefliyoruz. Çevrimdışı olarak bir davranış politikası öğrenmek üzere, İRE verilerini pekiştirmeli öğrenme için kullanan bir formülasyon öneriyoruz. Bu formülasyon, ödül olarak tanımladığı kullanıcı ilgisini enbüyükleyen bir arka kanal politikası bulmaya çalışır. Bu eniyileme probleminin çözümü için veri kümesindeki örneklerin gradyan güncellemesini kayıtlı veriden pekiştirmeli öğrenen üç farklı algoritma öneriyoruz. Bu algoritmaları değerlendirmek için bağımsız politika ile politika değerlendirme ve Bellman kalıntısı gibi çevrimdışı değerlendirme yöntemlerini kullanıyoruz. Tezde ayrıca, insansı bir yüz robotu kullanarak bir İRE kullanıcı deneyinin tasarımını yaptık. Bu etkileşim deneyi bir hikaye şekillendirme senaryosu altında, öğrenilen arka kanal politikasının kafa sallama ve gülümseme davranışlarını kontrol ettiği üç boyutlu bir yüz robotu ile gerçekleşir. Robotun sosyal davranışlarının katılımcılar üzerindeki etkisinin değerlendirilmesi öznel anket sonuçları ve etkileşimde hesaplanan ilgi seviyesi değerleriyle yapılmıştır. Kural tabanlı referans bir davranış politikası ile karşılaştırıldığında pekiştirmeli öğrenme politikasının daha kabul edilebilir olduğu, istatistiksel olarak anlamlı değerlendirme skorlarıyla gösterilmiştir. Bu tezde sunduğumuz çalışma, sosyal etkileşim robotlarında yalnızca bir tür robot davranışını ele alıyor olsa da, öncü bir çalışma olarak, İRE tasarımında kullanılabilecek alternatif metriklerin eniyilenmesini hedefleyen birçok başka robot davranış türünün otomasyonunun da yolunu açmaktadır.

ACKNOWLEDGMENTS

I express deep gratitude to my advisors Prof. Engin Erzin and Prof. Yücel Yemez for their guidance. I owe much to them for the long brainstorming sessions and the clarity they brought to the ideas. I am especially obliged to thank Assoc. Prof. Metin Sezgin for his insightful opinions and suggestions. Many thanks to Prof. Murat Tekalp for his feedback in his role as a member of my PhD committee. I am much grateful to my husband Talha for the immense support he gave me during my research, and to my son Hashim for all his love. They have remained my pillars of strength during numerous tough days. I am also thankful to my mother and father, and all my family for all the encouragement they have given me and the keen interest they have always taken in my education and my life. I thank my closest friends Bushra and Kiran with whom I have shared many good times during graduate school. Finally, I have to mention my amazing colleagues in our research lab. Buket, Ege, Tuğçe, Öykü, Berker, Akın, Ogün and Onur, because of all of you our lab was always full of laughs.

TABLE OF CONTENTS

List of Tables	x
List of Figures	xi
Abbreviations	xiii
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Problem Statement and Research Questions	2
1.3 Novelty and Contributions of the Work	4
1.4 Structure of the Thesis	6
1.5 Publications	9
Chapter 2: Recent Advancements on Engaging Social Robots	10
2.1 Introduction to Social Robots	10
2.2 What is Engagement?	10
2.3 Measuring Engagement	12
2.3.1 Subjectivity-oriented Methods	12
2.3.2 Objectivity-oriented Methods	13
2.3.3 Rich’s Connection Event Approach	14
2.4 Fostering User Engagement in HRI	16
2.4.1 Rule-based Techniques	17
2.4.2 Reinforcement Learning Methods	19
Chapter 3: Batch Reinforcement Learning: Concepts, Methods and Significance in Real-World Problems	21

3.1	Introduction	21
3.2	Markov Decision Process	23
3.3	Reinforcement Learning Framework	24
3.4	Reinforcement Learning Algorithms	26
3.5	Batch Reinforcement Learning	30
3.5.1	Concept	30
3.5.2	Challenges of Batch Reinforcement Learning	31
3.5.3	Example Applications	32
3.5.4	Existing Batch Learning Methods	37
3.6	Off-Policy Policy Evaluation Methods	38
Chapter 4:	MDP Formulation for Engaging HRI	41
4.1	MDP Formulation	41
4.1.1	State / Speech Features	41
4.1.2	Action / Backchannels	42
4.1.3	Reward / Engagement	42
4.2	Off-Policy Batch Data	43
4.3	Offline Evaluation Methods	44
4.3.1	Bellman Residual	44
4.3.2	Off-Policy Policy Evaluation	45
4.3.3	Similarity to Human Behavior	45
Chapter 5:	Proposed Algorithms	46
5.1	Introduction	46
5.2	Approach I: Batch-DQN for Policy Optimization in MDP	46
5.2.1	Algorithm	47
5.2.2	Offline Training	48
5.2.3	Offline Evaluation	49
5.3	Approach II: Deep Recurrent Q-Learning for Policy Optimization in POMDP	52

5.3.1	Algorithm	53
5.3.2	Offline Training	55
5.3.3	Offline Evaluation	55
5.4	Approach III: Sequential Random DQN with Constraint for Policy Optimization in MDP	58
5.4.1	Algorithm	59
5.4.2	Offline Training	61
5.4.3	Offline Evaluation	64
5.5	Discussion	65
Chapter 6:	Human-Robot Interaction Experiments	68
6.1	Experimental Design	69
6.1.1	Setup	69
6.1.2	Procedure	70
6.1.3	Story shaping game	71
6.1.4	Architecture	72
6.2	Experimental Results	76
6.2.1	User-Feedback Analysis	77
6.2.2	Engagement Analysis	79
Chapter 7:	Conclusion	81
7.1	Summary of the Thesis	81
7.2	Discussion	83
7.3	Future Work	84
	Bibliography	86

LIST OF TABLES

5.1	Batch-DQN: List of hyper-parameters and their values.	50
5.2	Batch-DQN: Bellman residuals for test set over 5-fold training	50
5.3	Batch-DQN: Off-policy policy evaluation (OPE) results	51
5.4	Batch-DQN: Similarity to human behavior.	52
5.5	DRQL: List of hyper-parameters and their values.	54
5.6	SRDQN: List of hyper-parameters and their values.	62
5.7	SRDQN: Off-policy Policy Evaluation results.	65
5.8	SRDQN: Similarity to human nods results.	66
5.9	SRDQN: Similarity to human smiles results.	66
6.1	Post-experiment questionnaire.	71
6.2	Example conversation between Furhat and two participants.	72
6.3	The p-values of each question in the post-experiment questionnaire using the paired T-test	79

LIST OF FIGURES

2.1	Timeline for directed gaze.	15
2.2	Timeline for mutual gaze connection event.. . . .	16
2.3	Timeline for adjacency pair connection event.	17
2.4	Timeline for backchannel connection event.	17
2.5	Engagement recognition module.	18
3.1	Categories of reinforcement learning formulation.	22
3.2	Reinforcement learning framework.	24
3.3	Actor-critic method	29
3.4	Classification of batch reinforcement learning.	31
3.5	Example of a recorded data for medical treatment applications. . . .	33
3.6	Data collection for offline training of robotic grasping task.	36
4.1	RL formulation of speech driven backchannel generation.	44
5.1	Bellman residual for batch-DQN.	51
5.2	DRQL: Bellman residual curves.	56
5.3	DRQL: Estimated Q-values.	57
5.4	DRQL: Off-policy policy evaluation (OPE) results.	58
5.5	SRDQN: Bellman residual and backchannel frequency.	63
6.1	Human-robot interaction setup.	68
6.2	Snapshots of human-robot interaction experiment.	69
6.3	System architecture for the HRI experiment.	73
6.4	Implementation of backchannel policy.	74

6.5	Backchannel event timing diagram for human-robot interaction experiment.	75
6.6	User evaluations in post-experiment questionnaire.	77
6.7	User engagement during the interaction and the corresponding connection events.	79
6.8	Cumulative engagement averaged over all participants	80



ABBREVIATIONS

Batch-DQN	Batch deep Q-network
Batch-RL	Batch reinforcement learning
DM	Direct methods
DP	Dynamic Programming
DQN	Deep Q-network
DR	Doubly robust
DRQL	Deep recurrent Q-learning
FC-LSTM	Fully connected LSTM
FQI	Fitted Q-iterations
HM	Hybrid methods
HRI	Human-robot interaction
IPS	Inverse Propensity Score
IS	Importance Sampling
LSTM	Long-short term memory
MDP	Markov decision process
MFCC	Mel-frequency cepstrum coefficients
MLP	Multi-layer perceptron
NFQ	Neural fitted Q-iterations
OPE	Off-policy policy evaluation
POMDP	Partially observable Markov decision process
RL	Reinforcement learning
SRDQN	Sequential random deep Q-network
WIS	Weighted importance Sampling
WDR	Weighted doubly robust

Chapter 1

INTRODUCTION

1.1 *Motivation*

Social robots are becoming more integrated into our daily lives, with an increasing prevalence in human environments like schools, museums and hospitals. Contrary to industrial or service robots, these robots work in close proximity to humans and are expected to display social behaviors that encourage their acceptance in the human company. The design of such robots targets to achieve a human-robot interaction (HRI) similar to a human-human interaction. Success is defined when humans engage in the interaction with likeability, comfort and naturalness.

A number of characteristics contribute to giving a social robot a human factor. These include advanced conversation skills, emotion recognition and display capabilities, user response based behavior adaptation, ability to develop a social relationship and the ability to display varying social roles [Fong et al., 2003]. A common metric targeted in the design of such features is the ability to sustain human engagement. Engagement is a common issue addressed by researchers and many use it to define the success of an interaction. An engaged user has higher chances of succeeding in the desired purpose of the interaction than a disengaged one. It is particularly important in interactive applications like tutoring, guidance in public places, assistance in work environment and aid in physical or psychological therapy.

The design of robots with engaging behaviors is a developing field and currently poses a number of open questions. For example, active research is being performed to understand the behaviors in a human-human interaction that contribute to sus-

tenance of mutual engagement. Another important line of work is the development of techniques and sensory cues that may measure and quantify engagement. However, the largest research gap, which is addressed by this thesis, is the automation of engaging behaviors in a social robot.

Several works have shown that a user perceives a robot as more human-like when it generates non-verbal facial and body gestures, along with strong verbal communication. This in turn increases the willingness of humans to engage with robots and strengthens the acceptability of robots into our daily lives. The common approach for automation of backchannel behavior in robots is rule-based methods. Supervised methods can also be applied if a dataset is created with optimal behavior demonstrations. However, these approaches do not bring a sense of purpose explicitly into the robot (i.e. to engage user). Moreover, the behavior learned from supervised learning cannot perform better than the reference behavior of the dataset. These works incorporate user's social signals to measure engagement and exploit it as the reward of the RL algorithm. A major issue faced when training with online reinforcement learning is that it requires direct interaction with humans. This is highly time-consuming and early interactions with an untrained robot can be frustrating. The main motivation of this thesis is the gap in literature of offline reinforcement learning methods for training a robot's non-verbal behaviors.

1.2 Problem Statement and Research Questions

In this thesis, we explore batch reinforcement learning (i.e. offline reinforcement learning) methods to train an agent non-verbal backchannels that maximize user engagement. We approach the problem using Markov decision process (MDP) to model the human-robot interaction. An MDP is a control system for decision-making problems where the process defines (discrete or continuous) states and emits rewards based on the action taken by a decision-maker. In our work, the robot is defined as the decision-maker and the interacting human as the process with states and rewards. We define rewards with user engagement so the optimal policy learns to

maximize engagement. In the design of the components of the MDP, the following questions need to be addressed:

- What audio-visual features are effective representations of user state? Does incorporating history into the formulation improve the performance?
- Which backchannel behaviors need to be optimized as discrete actions for the maximizing engagement?
- The reward is generated as a single scalar number from human engagement at each time step. What are the methods used to quantify engagement and how can we measure it in a real-time system?

As mentioned in section 1.1, the core challenge of the human-robot interaction problem is the difficulty in training with online reinforcement learning methods. Online training requires a robot to interact directly with humans, possibly over many days, and learn gradually from the responses (positive and negative) generated by the interacting persons. Offline training with batch reinforcement learning is an emerging field for numerous other real-world problems, where interaction with the environment using an untrained policy may be time-consuming, dangerous, costly or even illegal. It creates the possibility of learning the policy offline from a set of transitions collected by another behavior policy.

However, offline reinforcement learning comes with its own challenges. The following questions are addressed in this thesis when designing offline reinforcement learning algorithms for the human-robot interaction problem.

1. How to utilize the existing human-human interaction datasets to learn decision-policies with batch reinforcement learning methods?
2. What are the deficiencies in existing batch reinforcement learning methods when training for HRI? What novel methods can be introduced to improve the performance?

3. What methods may be used to evaluate the policy learned from the offline methods, prior to its the deployment into the real-world?

The final research target carried out by this thesis is the design of a user study in a human-robot interaction setup to evaluate the offline trained policy. The interaction scenario needs to be planned so that we may assess the impact of robot’s social behavior on the participants.

1.3 Novelty and Contributions of the Work

Our research explores techniques for training a robot’s non-verbal backchannels that improve engagement from the subjects during a human-robot interaction. The main contributions to this area can be summarized in the following points:

- **Formulation of the problem as Markov decision process.** We propose a Markov decision process (MDP) formulation for backchannel behavior that enhances engagement of the user during human-robot interaction. Conditional on the state of the human, the robot takes the decision of generating a backchannel event. The reward received from the subject reinforces the effectiveness of current policy. We describe the states from the subject’s speech features and rewards are calculated by his/her engagement levels.
- **Using human-to-human dyadic datasets as off-policy batch data.** We present the use of recorded human-to-human dyadic interaction datasets as a batch of trajectories acquired by a behavior policy (another human). The dataset is processed to extract speech features, actions and engagement at a time step. This data is then fed to a batch reinforcement learning (batch-RL) algorithm that uses neural networks to represent the Q-network and solves for the optimal backchannel policy.
- **Novel Batch-RL techniques** We propose three value-based off-policy batch-RL algorithms for training backchannel behavior models for robots in HRI.

The three algorithms, batch-DQN, deep recurrent Q-learning (DRQL), and sequential random deep Q-network (SRDQN), are characterized with respective strengths and limitations. Batch-DQN has advantage when training with large datasets, but assumes a fully-observable system. On the other hand, DRQL explicitly acknowledges the partial-observability of the problem and incorporates observation history with recurrent structures to infer about the true state information. However, it is hard to stabilize, and the training is computationally demanding and time-consuming. SRDQN aggregates the strengths of first two approaches and uses sequential information with a simpler network. Additionally, a constraint is introduced in the learning process to mitigate the harmful distributional shift problem.

- **Human-robot interaction experiments** A final major contribution of this thesis is the conduction of human-robot experiments with the policy that was evaluated best using offline metrics. An interaction setup was developed with the robot Furhat, and human subjects participated in a story-shaping game. During the interaction, the robot generated nods and smiles as backchannels following its RL-policy. Analysis of the user engagement and the feedback questionnaire filled by the participants proved the preference of the RL trained system over a baseline rule-based policy.

In a broader view, the proposed MDP formulation for the problem and its solution with the novel techniques may be generalized to any targeted robot behavior. We have shown here experiments for the non-verbal behaviors of nods and smiles, but this may be extended to other behaviors like appropriate eye gaze, correct display of emotions, verbal expressions like ‘hmms’ and ‘umms’ etc. Moreover, here we show how the engagement maximization can be explicitly incorporated into the formulation. On similar grounds other metrics of HRI like persuasiveness, trust, compliance [Steinfeld et al., 2006] can be used, or the problem can be modified as a multiple-objective task.

1.4 Structure of the Thesis

This document is organized into ten chapters. The first three chapters give an introduction to the thesis and provides a detailed literature review of our research domain. Chapter 4 to 8 are the core of the thesis which describe the problem formulation, offline training and evaluation methods, and online human-robot interaction user study. Chapter 9 summarizes the thesis work. Each one of the chapters is outlined below:

1. *Introduction*: This chapter starts with motivation behind the work and the research gap our work aims to fill. It then presents the problem statement and the research questions addressed here. At the end a brief description is presented on the novelties and contributions.
2. *Recent Advancements on Engaging Social Robots*: This chapter explains how social robots are getting integrated into our daily lives. It emphasizes user engagement as a key goal in the design of social robots. The chapter goes on to describe what is the technical definition of engagement and how researchers have proposed its measurement. Finally, the literature review is given on proposed methods for fostering user engagement in human-robot interaction, with focus around the works employing backchanneling behavior.
3. *Batch Reinforcement Learning: Concepts, Methods and Significance in Real-World Problems*: This chapter begins with the basic concepts of reinforcement learning (RL) and describes its formulation as a Markov decision process (MDP). The key components of the RL-framework are presented as well as the techniques utilized to solve MDPs. The discussion then moves to how the online reinforcement learning methods can be modified as offline techniques where interaction with the environment is no longer required. Several challenges are explained in the design of offline reinforcement learning or batch reinforcement learning. The significance of batch reinforcement learning is presented along with a brief description of the existing methods are briefly

described.

4. *Problem Formulation for Engaging HRI*: This chapter describes the formulation of the robot backchanneling problem as a Markov decision process (MDP). The key elements of the MDP are defined such that the solution to the MDP maximizes engagement. Furthermore, the use of IEMOCAP human-human interaction dataset as a collection of trajectories is illustrated. The chapter then presents details on the training steps with a batch learning algorithm. It concludes with a description of the offline evaluation metrics used later on in the thesis, for the evaluation of trained policies prior to their deployment into the real environment.
5. *Proposed Algorithms*: In this chapter we present three novel algorithms for the optimization of the robot's backchanneling policy to maximize user engagement. Assuming the state is fully observable, the novel Batch-DQN algorithm works iteratively to determine the solution from the fixed batch of decorrelated samples. The algorithm steps followed by batch-DQN are explained in detail. The training is performed for smiles as backchannels using the state-action transition data prepared from the human-human interaction dataset, and compared with existing batch reinforcement learning methods 'neural fitted Q-iteration (NFQ)'. The results demonstrate the potential of the proposed method to train a robot for engaging behaviors in an offline manner. In our second algorithm, we approach the problem as a partially observable MDP, where true state information is hidden and only observations are available. The deep recurrent Q-learning (DRQL) algorithm is designed with LSTMs in order to incorporate history of observation into the learning process. The details of the algorithm and how it helps to achieve the solution closer to the optimal are provided in this chapter. The training steps and the outcomes of offline evaluation metrics are discussed at the end of the chapter. The final algorithm combines the strengths of batch-DQN and DRQL. We address the questions: *How may we mitigate the distributional shift problem where*

the distribution of states visited during training differ from the distribution of states visited during testing? Can constraints be introduced in the learning process to bring the learned policy closer to the dataset policy? If we train for other backchannels, how do the results compare and contrast? The SRDQN algorithm is proposed which introduces the key idea of using trajectories (of sequentially related samples) to make each update. This is contrary to batch-DQN where an update is made from random decorrelated samples. Moreover, we introduce a **reward factor** (κ) during training, as a hyper-parameter to control the backchannel predictions by the policy and mitigate the distribution shift problem.

6. *Human-Robot Interaction Experiments:* The final part of the thesis work describes the procedure adopted to conduct a user study on human-robot interaction with a backchanneling robot. The interaction is designed with the robot *Furhat*, in a story-shaping interaction scenario. During the interaction, the robot generates nod and smile backchannels during listening turns, to increase the engagement of the participants. The experiment is evaluated subjectively as well as objectively. For the subjectivity-oriented evaluation, a questionnaire is provided to the participants after the interaction, to get feedback on their perception of the interaction in terms of engagement and naturalness. The objectivity-oriented evaluation is performed by annotating engagement indicating cues from video recordings of the experiment, and calculating engagement values.
7. *Conclusion:* The document concludes with this final chapter, where we summarize the main ideas and methods described throughout the thesis. It provides an insight into the contribution of the work presented here and its impact on future development of social robots. Moreover, this chapter discusses the open issues, describes limitations of the proposed model, and gives the main directions in order to address future improvements.

1.5 Publications

During the course of the research the following scientific publications are achieved from the work.

1. N. Hussain, E. Erzin, T. M. Sezgin, and Y. Yemez, “Speech driven backchannel generation using deep Q-Network for enhancing engagement in human-robot interaction”, Proc. Interspeech 2019, pp.4445–4449, 2019.
2. N. Hussain, E. Erzin, T. M. Sezgin, and Y. Yemez, “Batch recurrent Q-learning for backchannel generation towards engaging agents,” in 2019, 8th International Conference on Affective Computing and Intelligent Interaction (ACII), IEEE, 2019, pp. 1–7.
3. N. Hussain, E. Erzin, T. M. Sezgin, and Y. Yemez, “Training socially engaging robots: automation of backchannel behaviors with batch reinforcement learning,” IEEE Transactions on Affective Computing, 2020 [submitted].

Chapter 2

RECENT ADVANCEMENTS ON ENGAGING SOCIAL ROBOTS

2.1 Introduction to Social Robots

The function of social robots is to interact with humans on an emotional level. This type of interaction is grounded in visual and tactile perception no less than in verbal communication. Social robots is an interdisciplinary field of social sciences and robotic engineering. While the social scientists try to understand how social robots may be effectively employed with humans, the robotics community work towards advancement of technical capabilities of the robots [Campa, 2016]. Social robot is described as a physically embodied, autonomous agent that communicates through social cues, displays adaptive learning behaviour, and mimics various emotional states [Darling, 2012]. Social robots present great potential for a wide range of applications such as therapy of children with autism [Cabibihan et al., 2013, Zhang et al., 2019], assistive social robots in elderly care [Broekens et al., 2009, Robinson et al., 2014], robots in teaching and learning [Belpaeme et al., 2018, van den Berghe et al., 2019], guides in public places [Mubin et al., 2018], and assistance in work places [Leite et al., 2013]. The main theme of our work is to build intelligent robots that can also communicate with non-verbal social cues in order to enhance the interaction experience during an HRI.

2.2 What is Engagement?

Engagement is a key research area in human-robot interaction (HRI) and related fields including human-computer interaction (HCI), affective computing, artificial intelligence, learning systems, adaptive computing and social networking. An en-

gaged user is more likely to benefit from the interaction, for example learning a new skill or a information guide at the airport, and hence many researchers rate the success of the interaction with the level of engagement displayed by the user [O'Brien and Toms, 2008, Anzalone et al., 2015]. The concept of user engagement has been employed to understand user experience, motivate students, analyse interaction online, guide autonomous systems, and to appraise technologies for well-being, civil services, social networks, and more [Doherty and Doherty, 2018].

Despite of being a major theme of research and a universal goal in human-robot interaction, engagement is a difficult concept to define due to its multidimensional nature and the difficulty to measure it. The definition of engagement is found to be quite flexible in literature and can be found with varying nuances, generally dependent on the context [Glas and Pelachaud, 2015, Doherty and Doherty, 2018]. As provided by Peters et al., one way to define a taxonomy of literature for the definitions of engagement is to distinguish with the following two components: *attentional involvement* and *emotional involvement* [Peters et al., 2009]. Generally only one component is used to define engagement, however, definitions can also interleave the two components since attention is driven by emotions. In our work, we focus on the attentional involvement of the user, which is a more common method used to measure engagement during a HRI. The single most-cited definition of engagement for the attentional involvement is by Sidner et al. who define it as follows: *“The process by which two (or more) participants establish, maintain and end their perceived connection [Sidner and Dzikovska, 2002]. This process includes: initial contact, negotiating a collaboration, checking that other is still taking part in interaction, evaluating staying involved, and deciding when to end connection”*. On the other hand, the definition provided by Hall et al. concentrates more towards the emotional involvement, in particular empathy, and define engagement as *“the fostering of emotional involvement intending to create a coherent cognitive and emotional experience which results in empathetic relations between a user and a synthetic character”* [Hall et al., 2005].

Another distinction in literature is based on whether engagement is defined as one-sided or multi-sided. The definition stated by Sidner et al. [Sidner and Dzиковska, 2002] is categorized as multi-sided, while Poggi [Poggi, 2007] formulates engagement as one-sided with the following definition: “*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*”. Another major distinction provided by Bickmore et al. [Bickmore et al., 2010] differentiates short-term vs. long-term engagement. The former deals with user engagement in performing a task while interacting with the agent, whereas the latter implies much longer periods of interactions with the system and concerns the degree of involvement of the user over time. Since our work targets only user engagement, we define engagement as one-sided for short and long interactions.

2.3 Measuring Engagement

Much versatility is found in literature for the methods used to measure engagement. This is inline with the fact that a variety of approaches have been used to define user engagement. The measurement techniques can be categorized as *subjectivity-oriented* and *objectivity-oriented* [Doherty and Doherty, 2018]. Subjectivity-oriented methods target to understand perceptual and experiential aspects by user’s self-report. Objectivity-oriented infer about engagement from various observable features and behavior cues. While some have used a single cue, it is more common to measure engagement using several cues that arise over a certain time window [Peters et al., 2005]. Both techniques have been used in the work presented in this thesis. While only objective measurement of engagement is used for training with reinforcement learning, both objectivity-oriented and subjectivity-oriented techniques were used for the final evaluation during a human-robot interaction experiment.

2.3.1 Subjectivity-oriented Methods

Subjectivity-oriented methods include methods of self-report such as questionnaires, observations and interviews. The questionnaire method can easily be applied, and

can provide rich responses and detailed analysis. It may vary in the choice of time at which the questionnaire is applied: pre-experience, in-experience or post-experience. The questions typically use a Likert-scale measure for the users to assess their experience.

2.3.2 Objectivity-oriented Methods

Objectivity-oriented methods infer about engagement without human involvement, and use techniques like audio-visual analysis, psycho-physiological measures, and logging behavior and interaction. The advantages of this approach are limited disruption of interaction and reduced user burden.

Audio-visual analysis

The simplest objectivity-oriented methods adopt single variables as proxies of engagement, such as face location as detectors of disengagement like quitting the interaction [Bohus and Horvitz, 2014], and smiles and laughs as indicators of being engaged [Weber et al., 2018]. More complex and high level behaviors are also commonly applied for engagement measurement. For example, negative emotional state of the user has been considered as a strong indicator of disengagement in the interaction [Schmitt et al., 2010]. Litman et al. present the importance of detection and avoidance of frustration in a tutoring systems, which is strongly linked to the level of engagement displayed by students [Litman and Forbes-Riley, 2006].

Some researchers argue that engagement cannot be measured from single cues, but rather from several cues that arise over a certain time window [Peters et al., 2005]. These cues are derived from numerous verbal and non-verbal signals (acoustic component of speech, facial expressions, gesture, gaze and body posture). Ishii and Nakano measured conversational engagement between a Wizard-of-Oz-controlled virtual agent and a user, in the application of a mobile phone salesperson, by observing virtual agent's gaze, eye-tracking data, gestures and speech transcriptions, along with the user-reported judgements of user attitude [Ishii and Nakano, 2008].

It is observed that engaged users behave more cooperatively, paying attention to the object referred to by the agent [Ishii and Nakano, 2010]. In the work by Ritschel et al., multiple social signals are used to calculate engagement including head tilt and orientation, openness of the posture, lean posture and backchannels [Ritschel et al., 2017]. Monkaresi et al. used features from videos, heart rate, animation units, and local binary patterns in three orthogonal planes (LBP-TOP) to train an engagement classifier with supervised learning [Monkaresi et al., 2016].

Psycho-physiological measures

Another broad category of measures which has been used to infer engagement are psycho-physiological. Electrocardiography (ECG), heat flux (HF), and electroencephalography (EEG) measures have all been employed as proxies for engagement [Belle et al., 2011]. The features extracted from the signals produced by these methods enable a binary classification of engagement when trained on data captured while eight subjects view two 20-minute sets of 3 to 5-minute-long video clips, with one set deemed engaging, the other not [Belle et al., 2011]. This model achieves an accuracy higher than 90% accuracy. EEG measures have also been used to infer music engagement without interrupting the listening, where predictive models, based on EEG and motion-tracking data, achieve 67% engagement classification accuracy across-subjects and 85% accuracy within-subjects experience [Blankertz et al., 2010].

2.3.3 Rich's Connection Event Approach

A popular objectivity-oriented approach for measuring engagement using user's audio-video stream has been proposed in [Rich et al., 2010], where the authors propose an intricate engagement detection and measuring scheme for collaborative interactions between human and computer. We employ this method for engagement measurement in our work, which defines four types of events as engagement indicators, referred to as connection events (CEs). These are:

- **Directed gaze event.** In directed gaze, one person (the initiator) looks and optionally points at some object in the environment. The other person

(the responder) follows this by looking at the same object. This event is often synchronized with the initiator referring to the object(s) in speech. The shifting of the attention of the responder signals his intention for cooperation and the desirability in continuation of the interaction. Figure 2.1 gives the timeline for the sequence of events.

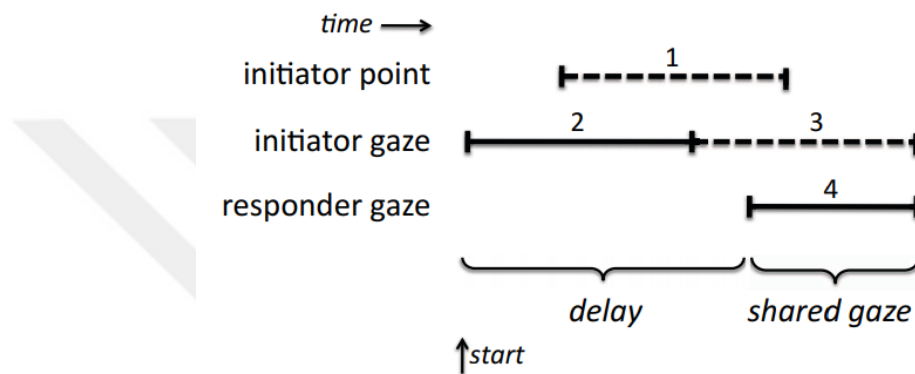


Figure 2.1: Timeline for directed gaze [Rich et al., 2010], with optional behaviors given by dotted line. (1) Initiator points to object. (2) Initiator starts to look at the object (begins before pointing). (3) Initiator may look away from the object to the responder, or continue the attention to the object. (4) Responder looks at the object.

- **Mutual facial gaze connection event..** The mutual gaze event occurs when an initiator looks at the other person, who responds back after some delay by looking back at the initiator. The delay can be zero if both parties look at each other simultaneously. Figure 2.2 shows the timeline for this event.
- **Adjacency pair.** In linguistics, provoking an utterance from the responder by the initiator is called an adjacency pair. It consists of minimal overlap or gap between the two speakers. The timeline for adjacency pair is shown in Figure 2.3.
- **Backchannel.** A backchannel event is where the responder gives feedback during the initiator's speaking turn with short verbal or non-verbal gestures. Typical examples of backchannels are smiles and expressions "hmm", "oh".

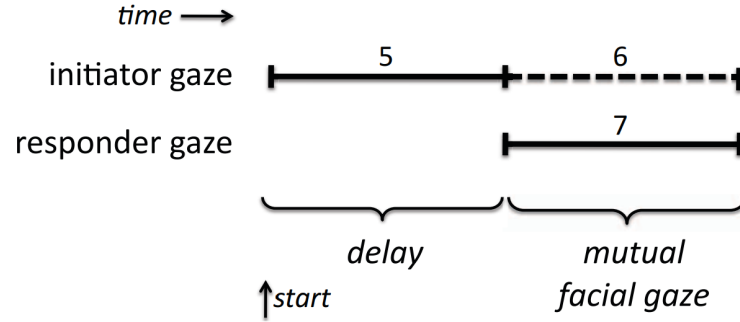


Figure 2.2: Timeline for mutual gaze [Rich et al., 2010], with optional behaviors given by dotted line. (5) Initiator looks at the responder (6,7) After some delay, responder looks at initiator and starts the mutual gaze period.

Unlike the other three connection event types, the start of a backchannel event is defined as the start of the responder’s behavior and this event has no concept of delay. Figure 2.4 shows the timeline.

The time between connection events is defined as the time between the start of successive events. To give a quantitative measure to engagement, the summarizing engagement metric called ‘mean time between connection events’ (MTBCE) is calculated from the connection events over a window of size T . MTBCE measures the frequency of successful connection events and is given by $MTBCE = \frac{T}{\text{no. of CEs in } T}$. Engagement is finally defined as the reciprocal of the MTBCE metric.

2.4 Fostering User Engagement in HRI

An engaging agent in addition to perceiving the user’s level of engagement should also be capable of maintaining it by exhibiting the appropriate socio-emotional behavior during the interaction. Two major issues arise regarding (1) the type of behavior to display and (2) when it should be exhibited. Providing backchannels as feedback is a key strategy for maintaining user engagement by providing agent’s listening behaviors [Lambertz, 2011]. Some example of backchannels are non-verbal gestures (e.g. head nods, smiles, facial expressions, eye gaze), non-verbal vocaliza-

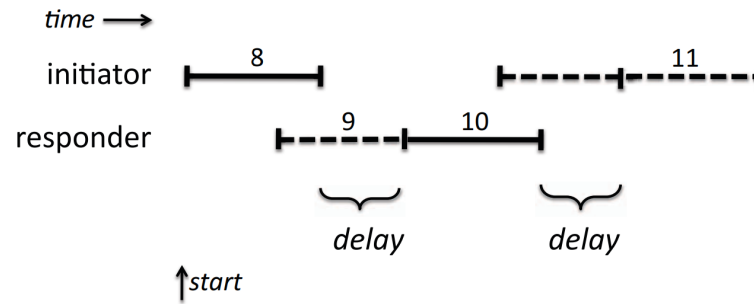


Figure 2.3: Timeline for adjacency pair [Rich et al., 2010], with optional behaviors given by dotted line. (8) *First turn* by the initiator. (9) The responder may take the *second turn* with delay zero and overlap with the first turn. (10) The responder takes over the turn with some delay. (11) A *third turn* may be followed by the initiator.



Figure 2.4: Timeline for backchannel event [Rich et al., 2010]. (12) Initiator performs the primary communication (13) Responder generates a brief verbal or gestural communication.

tions ('mm', 'uh-huh'), and verbal expressions ('yes', 'right'). These backchannels may be defined as non-intrusive social signals provided by the listener during the speaker's turn [Yngve, 1970]. Another key feature found to have direct impact on user engagement is the anthropomorphic embodiment of a robot. In this literature review, we restrict the discussion to the behaviors that influence user engagement.

2.4.1 Rule-based Techniques

Several researchers have used rule-based backchannel policies to study the impact they have on user engagement. The governing rules may be categorized as '*context-specific*' or '*context-independent*' approaches. In the '*context-specific*' approach, we

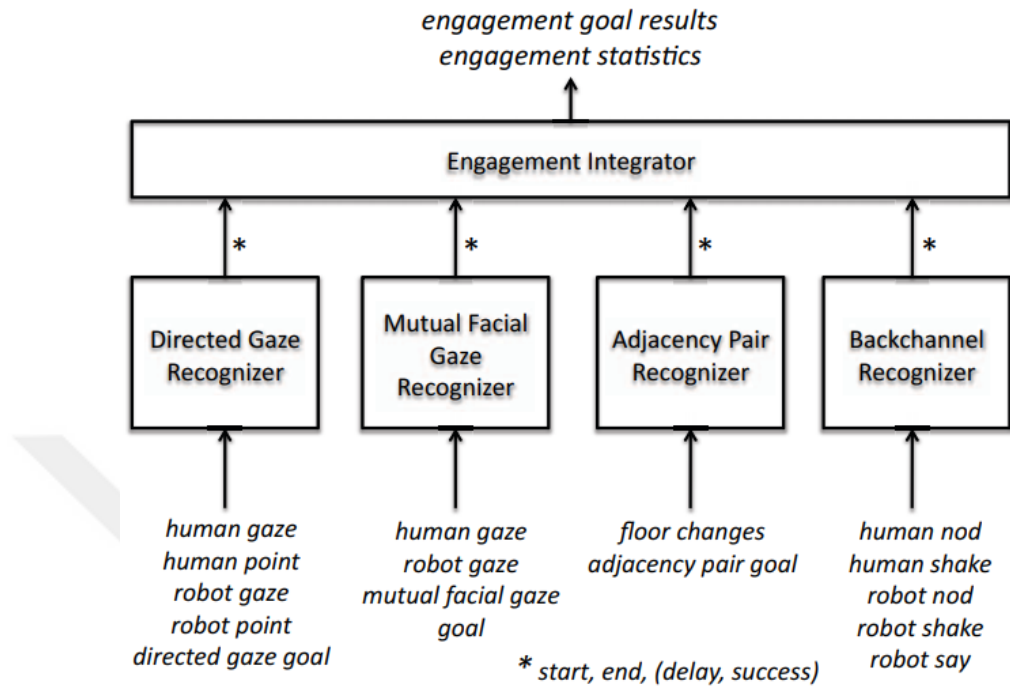


Figure 2.5: Architecture of engagement recognition module with the four types of Connection Events [Rich et al., 2010]

can find an example by Saerbeck et al., who conduct a language learning experiment with iCAT robot and show that motivational expressions like “don’t worry” and positive feedback to a correct answer with a nod and smile improve the learning performance of the students [Saerbeck et al., 2010]. Similarly, in another study with the DARWIN robot, various behavioral strategies are tested with students during a mathematics test conducted on the tablet device [Brown et al., 2013]. Their study shows the effectiveness of non-verbal feedback after each answer (agent nods at correct answer and shakes the head at incorrect answer), attention shift between student and tablet (to put emphasis when the student is addressed) and supportive phrases in improving the engagement and test performance of the students. The interaction experience of cognitively impaired seniors during an assistive activity with a robot with dynamic facial expressions and gestures is investigated [Moro et al., 2019]. Their results show that a human-like robot having an expressive face

and arm gestures significantly increases levels of engagement, positive affect, and perceived social intelligence during the interaction.

The existing works in the ‘context-independent’ category can be generalized and extended to other human-robot interaction scenarios. Leite et al. present such an example, and show that an emphatic model for backchannels conditional on the user’s affective state helps to keep the users engaged [Leite et al., 2014]. They conduct the experiment with the iCAT robot playing the chess game with children over long-term and used positive reinforcements from the robot with facial expressions and dialogue. Schroder et al. show that robots that generate multi-modal backchannels following simple probabilistic rules, are perceived as sensitive listeners and they are more competent in sustaining a conversational dialog [Schroder et al., 2011]. They demonstrate this with a rule-based strategy that a backchannel is triggered with a probability when the user nods or a variation is observed in the pitch of the user’s voice. In another work, the prosody and pause behaviors of the user are used to construct a rule-based backchannel policy [Truong et al., 2010].

2.4.2 Reinforcement Learning Methods

In HRI, different aspects of interaction have been addressed with reinforcement learning, for example learning empathetic supportive strategies [Leite et al., 2011], affective behavior of a tutoring robot [Gordon et al., 2016], and adapting personality in the assistive domain for post-stroke users [Tapus et al., 2008] or children with autism [Liu et al., 2008]. Qureshi et al. train a robot for the social skill of greeting via RL and reward it whenever a person successfully performs a handshake [Qureshi et al., 2016, Qureshi et al., 2017]. Robot gaze control has also been a target of interest with reinforcement learning for some researchers [Mitsunaga et al., 2006, Lathuilière et al., 2019].

While RL is a popular learning framework in HRI, the research works that incorporate engagement and related social signals as rewards are still scarce. There exist some works in the literature, which aim to enhance engagement in human-robot

interaction by using online reinforcement learning (RL). These works incorporate user's social signals to measure user engagement and exploit it as the reward of the RL algorithm. A versatile range of behavior models can be found, that have been trained and optimized to maximize engagement. Ritschel et al. use online reinforcement learning to adapt the personality of a robot, by varying extroversion through linguistic styles, with the goal of keeping the user engaged [Ritschel et al., 2017]. By using social signals like head tilt and openness of the body to estimate engagement, they focus on using engagement as reward and feedback for state representation, in a storytelling scenario. Weber et al. adapt the robot's 'sense of humor' to the user's preference as an approach to bonding and engagement [Weber et al., 2018]. The reward signal is defined from user's smiles and laughs as indicators of engagement. Gordon et al. exploit facial expressions to measure child's engagement, in a language tutoring setup, in order to adapt a robot's behaviors [Gordon et al., 2016]. They define 24 discrete states conditioned on valence and engagement of the student and train with RL to maximize rewards derived from valence and engagement. Liu et al. use RL in the context of a basketball game for children with autism spectrum disorder [Liu et al., 2008]. They adapt the robot's behavior to the child's affective state and estimate child's 'liking' based on physiological signals as indicator of engagement and use it as a reward function.

The works cited above have used reinforcement learning for a variety of behaviors to foster engagement. However, no works yet exist for optimizing non-verbal backchannels that explicitly maximize engagement. This observation is the key motivation behind the study presented in this thesis.

Chapter 3

BATCH REINFORCEMENT LEARNING: CONCEPTS, METHODS AND SIGNIFICANCE IN REAL-WORLD PROBLEMS

3.1 Introduction

Reinforcement learning provides a mathematical formalism for sequential decision making problems. It trains for optimal action decisions which maximize some notion of rewards. In the past decade, the introduction of deep neural networks as value function approximators, along with advancements in the training algorithms has resulted in pronounced success of reinforcement learning methods in a wide range of domains. While early success of reinforcement learning has been demonstrated in gaming domains like backgammon [Tesauro, 1994], Atari [Mnih et al., 2015] and Go [Silver et al., 2017], researchers are now intrigued in exploring its effectiveness in numerous real-world problems like robotics [Levine et al., 2018a, Kahn et al., 2018], medicine [Raghu et al., 2017b, Wang et al., 2018] and self-driving cars [Sallab et al., 2017, Pan et al., 2017a].

However, the fundamental online learning paradigm of the reinforcement learning methods hinders its wide-spread. The online techniques follow the iterative process of interaction with the environment, either with latest learned policy (on-policy methods) or another behavior policy (off-policy), and policy update from the experience [Sutton and Barto, 2018]. For many settings in real-world, this kind of approach presents numerous difficulties and becomes impractical. The foremost challenge is the incapability to deploy an unknown or an untrained random policy into a real-world environment. Such a practice may violate many safety constraints

and may be dangerous (self-driving cars), costly (robotics) or even illegal (health applications). Similarly, in our work on human-robot interaction applications, training sessions with a human over several days is highly cumbersome and not practical. Furthermore, in online learning, the latest experience is used to update the policy, but then it is immediately forgotten. This means that the data from the earlier visited states cannot be re-utilized. Works have shown that re-visiting past history and its repeated utilization in policy update boosts the learning process [Ernst et al., 2005, Mnih et al., 2015, Hester et al., 2018].

A solution to this problem is the use of batch reinforcement learning (batch-RL) methods, that encourage first the collection of data and then learning of the policy from this batch data in an offline manner [Ernst et al., 2005, Lange et al., 2012]. The collected data is generally in the form of set of transitions sampled from the environment and the crucial task of batch reinforcement learning algorithms is to extract the best out of the fixed batch of samples. In a pure batch reinforcement learning paradigm, the batch of samples is collected once and the agent does not interact with the environment during any step of the training. This is also commonly known as offline reinforcement learning [Levine et al., 2020]. Figure 3.1 illustrates the different formalism of reinforcement learning depending on the method used to collect the samples and update the policy.

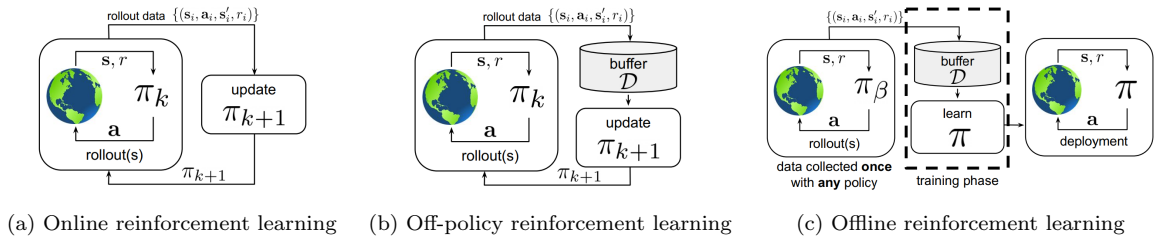


Figure 3.1: Reinforcement learning formulation as (a) Online reinforcement learning (b) Off-policy reinforcement learning (c) Offline reinforcement learning [Levine et al., 2020]

In the classical online reinforcement learning, Figure 3.1(a), the policy π_k is

updated from samples collected by itself during repeated interaction with the environment. In a typical off-policy reinforcement learning setting, new samples are collected by the policy π_k and appended to an experience replay buffer $\mathcal{D}$, such that $\mathcal{D}$ is composed of samples from $\pi_1, \pi_2, \dots, \pi_k$. Each update of the policy uses the entire history of experience to learn the new policy π_{k+1} . Offline reinforcement learning typically keeps the data collection part as a single step process. A behavior policy interacts with the environment over a long time and creates an experience replay buffer of samples. The training process does not interact with the MDP at all, and the policy is only deployed after being fully trained.

In this chapter, we cover the formulation of reinforcement learning as a Markov decision process and discuss the conceptual approaches used to optimize the solution. We then provide the reader with the tools of batch reinforcement learning and describe the challenges faced in this setting. Following this we describe some of the existing batch reinforcement learning techniques available in literature and present the applications that have been explored despite the limitations of the methods.

3.2 Markov Decision Process

In general, the reinforcement learning formulates the optimization problem as a Markov decision process (MDP), which is a mathematical formulation for discrete-time stochastic control problem. MDP is defined as tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, d_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is action space, $\mathcal{P}$ is the state transition probability function, $\mathcal{R}$ is the reward function, d_0 is the initial state distribution function, and $\gamma \in [0, 1)$ [Sutton and Barto, 2018]. In an MDP, at each time step t , the process is in some state $s_t \sim \mathcal{S}$ and a decision policy takes an action $a_t \sim \mathcal{A}$. The transition to next state s_{t+1} is partly random and partly controlled by the decision policy, defined by the state transition function $\mathcal{P}(s_{t+1}|s_t, a_t)$. Furthermore, the process responds to each action of the decision maker by generating a reward $r_t \sim \mathcal{R}(s_t, a_t)$. The discount factor γ weighs the future rewards, determining the temporal extent of the action's effects. Markov decision process (MDP) is an extension to Markov chain

with an addition of choice of action and motivation for the action via rewards. The solution to any MDP is a policy $\pi^*(a|s)$ which maximizes the expectation of sum of discounted rewards: $\pi^* = \operatorname{argmax}_{\pi} E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots | s_t = s, a_t = a, \pi]$.

A partially observed Markov decision process (POMDP) is defined where true state information is not available, and only observations are emitted with some distribution over the states. A POMDP is defined as the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \mathcal{E}, d_0, \gamma)$, where the same definitions hold for $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, d_0$ and γ , $\mathcal{O}$ is the observation space and $\mathcal{E}$ is the emission function. At time t , the observation $o_t \sim \mathcal{O}$ is observed sampled from the distribution defined by the emission function $\mathcal{E}(o_t|s_t)$.

3.3 Reinforcement Learning Framework

The formalism of reinforcement learning as a Markov decision process is shown in Figure 3.2. The RL methods define an environment for the agent to interact with continuously. The agent learns by exploring the effect of its actions on the state transitions and the rewards emitted by the environment. It understands the actions that may maximize the cumulative rewards and hence optimizes the decision policy by learning over time [Brewka, 1996].

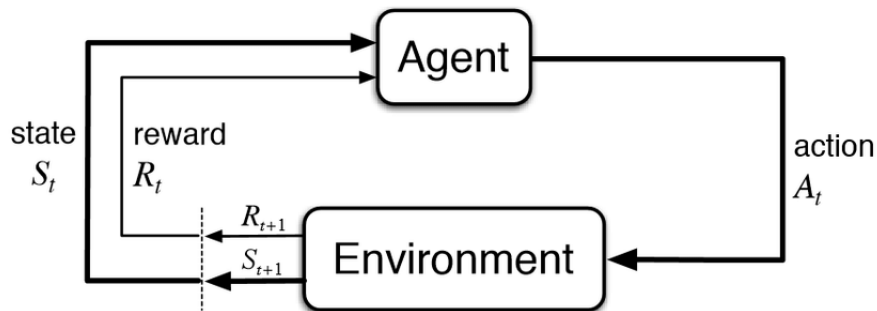


Figure 3.2: The agent–environment interaction in reinforcement learning [Sutton and Barto, 2018])

Beyond environment and agent, four sub-elements of reinforcement learning may

be identified: a policy, reward signal, a value function, and, optionally, a model of the environment [Sutton and Barto, 2018].

- **Policy:** A policy $\pi(a_t|s_t)$ defines the distribution of the agent's behavior conditioned on the current state of the environment, or conditioned on the observation $\pi(a_t|o_t)$ in a partially observable Markov decision process. The policy may also be conditioned over a history of observations $\pi(a_t|o_{0:t})$. It may take the form of a simple look-up table or a more complex function with non-linear equations and deep neural networks. The core of reinforcement learning is to learn the optimal policy that maximizes the cumulative discounted rewards.
- **Rewards Signal:** The goal of the reinforcement learning task is defined by the reward signal generated by the environment. Given a state s , a reward signal $r(s, a)$ is a distribution dependent on the state and the action. A high reward gives greater credit to the action taken while a low reward discredits it.
- **Value Function:** The state value function $V_\pi(s)$ defines how good a particular state s is for a the policy π . It estimates expected cumulative reward that is obtained by following the policy π when starting the state s till the end of episode. The state-action value function $Q_\pi(s, a)$ estimates the value when starting from the state-action tuple (s, a) . These value functions are given as:

$$V_\pi(s) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}) | s_t = s \right] \quad (3.1)$$

$$Q_\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}) | s_t = s, a_t = a \right] \quad (3.2)$$

Where the reward determines the immediate, intrinsic desirability of the environmental state, the values indicate the long-term desirability of the state after taking into account the states that are likely to follow, and the rewards available in those states. The optimum solution to a reinforcement learning

problem results in a value function $V^*(s) = \max_{\pi} V_{\pi}(s)$, where the policy takes actions that achieve maximum rewards over the long run.

- **Model of the Environment:** Model is the agent's perception of the environment. It allows inference to be made about how the environment will make state transitions and what rewards will be generated for the actions given a certain state. Methods that use models for solving the RL problems are called *model-based methods* and methods that do not require a model and are explicitly trial and error methods are called *model-free methods*.

A trajectory followed by the policy π is defined as the sequence of states and actions, given by $\tau = (s_0, a_0, s_1, a_1, \dots, s_H, a_H)$, where H is the length of trajectory that may be finite or infinite. The trajectory distribution may be defined from this, for a given MDP $\mathcal{M}$ and policy π , as:

$$p_{\pi}(\tau) = d_0(s_0) \prod_{t=0}^H \pi(a_t|s_t) P(s_{t+1}|s_t, a_t). \quad (3.3)$$

The reinforcement learning objective under the trajectory distribution is defined to be:

$$J(\pi) = E_{\tau \sim p_{\pi}(\tau)} \left[\sum_{t=0}^H \gamma^t r(s_t, a_t) \right]. \quad (3.4)$$

3.4 Reinforcement Learning Algorithms

In this section we describe the four main categories of existing reinforcement learning methods. In principle, any method from these categories may be approached as a batch reinforcement learning problem but not all are effective in the offline setting. These issues are discussed in section 3.5. The concept behind any reinforcement learning algorithm is identical where they follow the same learning loop: the agent *interacts* with the environment following either its own policy $\pi(a_t|s_t)$ or some behaviour policy $\pi_b(a_t|s_t)$ and uses the observed transition (s_t, a_t, r_t, s_{t+1}) to *update* its policy. Methods that use a behavior policy to collect samples are called ‘off-policy’ reinforcement learning methods. The agent may also store a history of

transitions in a buffer $\mathcal{D} = \{(s_t^i, a_t^i, r_t^i, s_{t+1}^i)\}$, and utilize the entire or part of the history to make the update. Without going into details, the four types of approaches are summarized below.

Policy gradients

Policy representation with deep neural networks implies the policy is generally parameterized with a parameter vector θ , and is given as $\pi_\theta(a_t|s_t)$. A direct way to optimize the objective function J given by equation 3.4 is to determine the gradients and update the weights in the direction to maximize the objective function. For a network that outputs the logits for the discrete actions a_t , the gradient of the objective function be given by:

$$\nabla_\theta J(\pi_\theta) = E_{\tau \sim p_\pi(\tau)} \sum_{t=0}^H \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) \underbrace{\left(\sum_{t'=t}^H \gamma^{t'-t} r(s_{t'}, a_{t'}) - b(s_t) \right)}_{\text{return estimate } \hat{A}(s_t, a_t)}. \quad (3.5)$$

The return estimator $\hat{A}(s_t, a_t)$ can itself be estimated with Monte Carlo samples, where samples are generated from $p_\pi(\tau)$ and the rewards are summed up over all the time steps. Another approach is to use neural network to estimate the return estimator which formulates the actor-critic methods. The baseline $b(s_t)$ can be estimated as the average reward over the sampled trajectories, or by using a value function estimator $V(s_t)$. For detailed reading on policy gradient methods, we refer the reader to the prior works ([Sutton and Barto, 2018, Kakade, 2002, Schulman et al., 2015]).

Approximate dynamic programming

Another popular approach is to follow a recursive process to learn the optimum state value function $V^*(s)$ or state-action value function $Q^*(s, a)$. The dynamic programming (DP) algorithms are obtained by turning the Bellman optimality equation for $V^*(s_t)$ (Equation 3.6) or $Q^*(s_t, a_t)$ (Equation 3.7), into update rules for improving

approximations of the desired value functions. A greedy policy can be derived from the state-action value function as $\pi(a_t|s_t) = \delta(a_t = \operatorname{argmax} Q(s_t, a_t))$.

$$V^*(s_t) = \max_{a_t} \sum_{s_{t+1}, r_t} [r_t + \gamma V^*(s_{t+1})] P(s_{t+1}, r_t | s_t, a_t), \quad (3.6)$$

$$Q^*(s_t, a_t) = \sum_{s_{t+1}, r_t} \left[r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) \right] P(s_{t+1}, r_t | s_t, a_t), \quad (3.7)$$

Based on this concept, Q-learning and actor-critic methods are the two commonly used algorithms. Q-Learning is an off-policy DP algorithm for estimating the long-term expected return i.e. the Q-values. It follows the iterative process of fitting the Bellman control equation [Sutton and Barto, 2018] given by:

$$Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \quad (3.8)$$

In order to utilize this equation for learning, the error can be defined by the difference between the left-hand side and right-hand side of Equation 3.8. Given that the loss function is defined as the mean square error, the parameter updates for the Q-function $Q_\phi(s_t, a_t)$, with weight ϕ , are made in the direction to minimize the loss function. Several variants of Q-learning exist, that differ in the length of history taken when minimizing the error. While the fitted Q-iteration [Ernst et al., 2005] fully minimize the error for entire history of samples, the original Q-learning [Watkins and Dayan, 1992] method takes a single gradient step. The commonly used variant in deep reinforcement learning is deep Q-network (DQN) [Mnih et al., 2015], that is a hybrid of the two methods and employs a replay buffer [Lin, 1992] for taking gradient steps on the Bellman error objective concurrently with data collection. The replay buffer is used to store a past history of data. Transitions are sampled randomly from the buffer to break the sequential correlation between the samples. Another concept used by DQN is the initialization of a second Q-network towards which the Q-values are iteratively updated. This target network is identical to the policy Q-network but the weights are refreshed periodically after a delay.

Actor-critic algorithms

The basic concepts from policy gradients are combined with approximate dynamic programming to formulate the actor-critic algorithms. The value function Q-network is utilized to better estimate the advantage function $\hat{A}(s, a)$, introduced in policy gradients (Section 3.4), for policy gradient calculation. The different variants of actor-critic methods can be categorized as on-policy [Konda and Tsitsiklis, 2000] and off-policy [Haarnoja et al., 2018, Heess et al., 2015]. The off-policy algorithms are of particular interest for us, since they can easily be extended to the offline setting. While Q-learning directly learns the optimal Q-function, the actor-critic methods target to learn the Q-function that corresponds to the current policy $\pi_\theta(a|s)$.

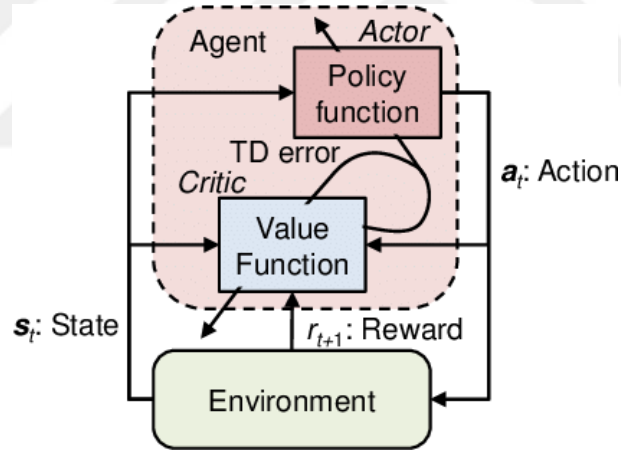


Figure 3.3: Actor-critic method

Model-based reinforcement learning

Several variants of model-based reinforcement learning algorithms exist. But in general, it refers to a broad class of methods that explicitly learn the transition dynamics $\mathcal{P}(s_{t+1}|s_t, a_t)$ and utilize it in the learning. When parameterized with ψ , the transition dynamics is represented as $\mathcal{P}_\psi(s_{t+1}|s_t, a_t)$. Some commonly used model-based RL methods utilize the learned transition dynamics model for planning

at test time, with various trajectory optimization methods [Nagabandi et al., 2018, Chua et al., 2018]. In addition to the transition dynamics, other methods also utilize the learned policy $\pi_\theta(a_t|s_t)$ and back-propagate through time to optimize the policy [Deisenroth and Rasmussen, 2011]. Another popular approach is to use the transition model to generate *synthetic* samples and augment them with model-free reinforcement learning algorithms including Q-learning [Sutton, 1991], policy gradients [Parmas et al., 2018, Kaiser et al., 2019] and actor-critic [Janner et al., 2019].

3.5 Batch Reinforcement Learning

3.5.1 Concept

Batch reinforcement learning is a class of methods where the reinforcement learning objective is optimized with a set of transitions sampled from the environment. In an early work, it was referred to as a fixed set of transitions and given a priori [Ernst et al., 2005]. However, to make distinction between *online*, *offline*, *batch* and *semi-batch*, Lange et. al. classified the batch algorithms from two perspectives: data-usage perspective and interaction perspective. Figure 3.4 give a pictorial representation of the classification of batch vs non-batch algorithms. A pure batch reinforcement learning algorithm, also referred to as offline reinforcement learning, collects the samples from the environment only once. A pure online algorithm returns to the environment for every sample and discards it immediately after its use in policy update. In between lie algorithms that store the past experience but keep returning back to the environment collect more data, to either refresh the experience buffer (semi-batch) or to append to the existing data (growing-batch) .

In this thesis, we used methods that work with pre-collected data and the agent does not interact with the environment to collect additional data. Therefore, we discuss here only offline reinforcement learning that may also be defined as *data-driven* reinforcement learning. The goal of offline reinforcement learning is to extract the best policy from the static dataset of transitions $\mathcal{D} = \{(s_t^i, a_t^i, r_t^i, s_{t+1}^i)\}$. We will

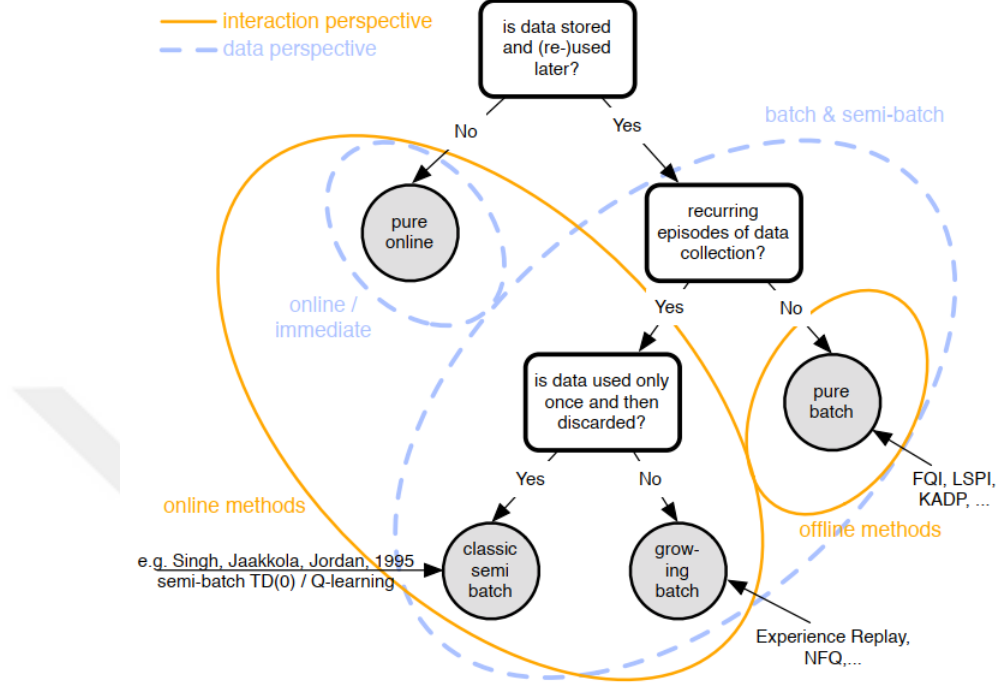


Figure 3.4: Classification of batch vs non-batch algorithms from two perspectives: interaction perspective and data-usage perspective [Lange et al., 2012]

use $\pi_{\mathcal{B}}$ to represent the behavior policy used to create the dataset, which determines the distribution of states sampled according to $s \sim d^{\pi_{\mathcal{B}}}(s)$ and actions sampled from $\pi_{\mathcal{B}}(a|s)$.

3.5.2 Challenges of Batch Reinforcement Learning

Offline reinforcement learning is a difficult problem and come with a number of challenges. The most obvious challenge is the incapability to improve exploration. The learning is restricted to the samples available in the batch data $\mathcal{D}$, which means there might be a possibility of state-action pair that result in higher rewards. For this reason, the optimum policy learned by offline reinforcement learning is limited to and is dependent on the batch, and cannot be generalized to the global optimum policy.

Another practical challenge comes from the core of reinforcement learning where it forms hypothesis about the counterfactual queries, that is, what might happen if the agent takes an alternate action from the one available in the data. This is inevitable in offline reinforcement learning since it targets to improve the policy performance relative to the behavior seen in the dataset $\mathcal{D}$. Contrary to this, standard supervised learning methods are designed on the assumption that the test data comes from the same distribution as the training data. The fundamental challenge of making counterfactual queries results in the problem of distributional shift where the function approximators are trained under one distribution from $d^{\pi_B}(s)$ and evaluation is performed under a different distribution $d^{\pi}(s)$ due to the goal of maximizing the expected return. To understand in depth the harm caused by distributional shift and how linear error bounds are converted to quadratic error bounds, we recommend the reading of some works [Ross et al., 2011, Levine et al., 2020].

One approach to address distributional shift is by bounding the shift so the learned policy is constrained to be close to the behavior policy [Kakade and Langford, 2002, Schulman et al., 2015]. Such techniques may be broadly classified as **policy constrained** methods. Another class of techniques is **uncertainty-based** methods which attempt to estimate the epistemic uncertainty of Q-values, and then utilize this uncertainty to detect distributional shift [Auer et al., 2009, Osband et al., 2016, O’Donoghue et al., 2018]. The epistemic distribution is substantially larger for out-of-distribution actions, hence can be used to produce conservative targets.

3.5.3 Example Applications

We can understand the significance and necessity of batch reinforcement learning methods from the example scenarios where only these methods are applicable. Such scenarios guide the design the batch algorithms to allow them to work well in practice.

Healthcare. In healthcare, safety is the first priority when opting for training

methods. For this reason, any possibility of online training is largely precluded. The medical illness of a patient is modelled with a Markov decision process. The optimum policy is decision about the course of interventions, like diagnostic tests and prescriptions, that will treat the patient. The true state information, defined by the condition of the patient, is hidden and can only be inferred from observations. The observations are generated from the patients symptoms and test results. A partially observable MDP formulation is most suitable for such cases.

Deployment of an untrained policy, or even a partially trained one, may result in serious outcomes. Offline training on treatment history of patients by their physicians may be the only viable solution. MIMIC-III is an important dataset containing medical records from ICUs [Johnson et al., 2016], that has influenced data-driven research on medical treatments. The approximately 60 thousand medical records of the dataset have been used in the treatment of sepsis [Raghu et al., 2017a] and optimized use of ventilators [Prasad et al., 2017] with Q-learning methods. The actor-critic methods have been applied by [Wang et al., 2018] on MIMIC-III to determine drug recommendations.

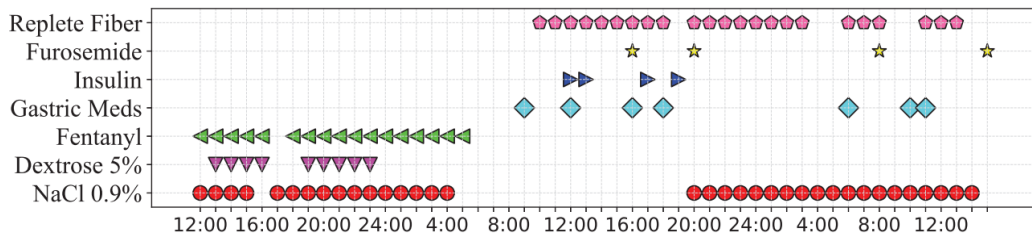


Figure 3.5: A record in MIMIC-3 showing a treatment steps of a patient with chronic kidney disease, acute respiratory failure, etc, when seven medications are randomly selected. The admission time 12:00 to the discharge time 17:00 is represented on the x-axis for the third day. Each prescribed medication is represented a unique icon. [Wang et al., 2018]

Offline RL outside ICU includes the use of fitted Q-iteration on data collected

from clinical trial involving 1460 patients to learn the optimum treatment options for schizophrenia [Shortreed et al., 2011]. Deep reinforcement learning (DRL) based on historical treatment plans was investigated by Tseng et al., for developing automated radiation adaptation protocols for lung cancer with model-based approach [Tseng et al., 2017]. Nie et al. studied the problem of learning dynamic when-to-treat policies with the advantage doubly robust estimator on prior data [Nie et al., 2019]. Guez et al. worked on finding optimal stimulation frequencies for reducing epileptic seizures optimal from recordings of seizure activity in mouse brains [Guez et al., 2008].

Self-driving cars. The task of self-driving car can be formulated as a reinforcement learning problem, such that the Markov decision process models the scene around the vehicle, actions are defined by car maneuvering and observation comes from the location of surrounding vehicles and the turns and bends of the road. This real-world application, however, also faces the challenge of safety similar to health-care domain. In online setting, exploration allows mistakes to happen so that the agent can learn from its negative outcome. This cannot be tolerated on the road where even a small mistake may endanger the lives of the passengers. One way to approach is to train in the real world with human interventions, in case the vehicle violates a safety constraint [Kendall et al., 2019]. Another preferred approach is the use of simulators to train the model [Sallab et al., 2017].

Batch reinforcement learning is still in the early stages of development for self-driving cars [Yurtsever et al., 2020]. Nevertheless, the popularity of data-driven learning approaches is constantly rising using large datasets containing thousands of hours of recorded real-live driving activity. RobotCar [Maddern et al., 2017] and BDD- 100K [Yu et al., 2018] are two important datasets that are commonly applied for training self-driving cars. The common offline approach for training autonomous driving is through imitation learning [Sun et al., 2018, Bojarski et al., 2016, Pan et al., 2017b, Codevilla et al., 2018]. The elements of imitation learning can also be provided with constrained model-based RL methods when training with

demonstration data, as for example in the case of deep imitative models (DIMs) [Rhinehart et al., 2018].

Goal-directed dialogue policies. Dialogue is an interactive sequential decision-making problem and can be modeled as an MDP especially when it is goal oriented. An example is a chat bot on any website which is providing information to the customer about the product, with the primary goal of persuading the customer to make the purchase. Since the goal for such robot is to interact with humans successfully, large number of live human interaction is required to train it. In such a case, data collection is a time consuming and expensive process. However, data collected offline directly from human conversation is more natural and easier to obtain.

Previously Zhou et al. have used offline RL approaches in the areas of dialogue and language interface [Zhou et al., 2017]. In their work dataset consisted of logged interactions such as agent-customer transcript. Dialogue management, which is typically concerned with accomplishing a specific goal such as retrieving information or filling forms, is an example of such a system. Various researchers have shown the use of offline RL to tackle the problems like restaurant information retrieval [Pietquin et al., 2011] flight booking [Henderson et al., 2008], and restaurant recommendation [Kandasamy et al., 2017]). Jaques et al. applied offline RL to the problem of dialogue generation [Jaques et al., 2019]. The focus of their research is to produce natural responses that produce positive feedback from human users.

Robotic manipulation skills

Offline reinforcement learning for robotics is appealing since a set of complex behaviour skills need to be acquired by a robot and repeated interaction with the environment is expensive. Collecting data once for multiple control policies, especially concerning raw sensory data like camera images, is far more practical. Robot visual perception alone requires large training sets, and hence necessitates the learning from datasets such as the ImageNet Large-Scale Visual Recognition Challenge [Russakovsky et al., 2015] containing a training set of 1.5 million images for open world object recognition.



Figure 3.6: Large-scale robotic grasping data collection. [Kalashnikov et al., 2018] describes how a dataset of over 500,000 grasp trials collected from multiple robots was used to train a vision-based grasping policy, comparing fully offline training and online fine-tuning.

The robotic grasping task is specially an appealing application, which has been explored by several prior works with offline reinforcement learning [Pinto and Gupta, 2016, Kalashnikov et al., 2018, Levine et al., 2018b]. Kalashnikov et al. used a dataset of 500,000 grasp trial to learn a vision-based policy with approximate dynamic programming techniques [Kalashnikov et al., 2018]. Pinto et al. utilized domain-specific algorithms, such as a single-step bandit formulation and used 50k grasp trials to train the robot [Pinto and Gupta, 2016].

Besides the grasping task, a model-based algorithm is proposed by Ebert et al., which uses predictions of future video frames to learn a variety of robotic skills with offline data [Ebert et al., 2018]. This approach is expanded later by a larger and more diverse dataset [Dasari et al., 2019]. In combination with offline RL, Cabi et al. proposed a user-friendly method for controlling robots by learning rewards from human preferences [Cabi et al., 2019]. For training the robot navigation task with offline RL, a dataset of visual indoor scenes collected via a camera-mounted-robot has been created [Mo et al., 2018]. Kahn et al. discuss how a method based on prediction of future reward signals, blending elements of model-based and model-free

learning, can learn effective navigation policies from data collected in the real world using a random exploration policy [Kahn et al., 2018]. Pure model-based methods in robotics typically involve training a model on real or simulated data, and then planning within the model to produce a policy that is executed on a real system. Approaches have included using Gaussian process models for controlling blimps (Ko et al., 2007), and using linear regression (Koppejan and Whiteson, 2009) and locally-weighted Bayesian regression (Bagnell and Schneider, 2001) for helicopter control.

3.5.4 Existing Batch Learning Methods

Dynamic programming methods, such as Q-learning, in principle can be directly used as offline reinforcement learning. These methods are utilized in this thesis for offline training with datasets. Earlier works are based on linear function approximation for Q-network such that $Q_\phi \approx f(s, a)^T \phi$ where $f(s, a)$ is a feature vector associated with the state-action pair [Lange et al., 2012, Lagoudakis and Parr, 2003]. When linear function approximation is used to represent the Q-function, the Q-function for a policy can be represented as the solution of a linear system, and hence can be computed via least squares minimization over the static dataset, since the Bellman operator is linear. Two of the popular methods, described in detail in [Sutton and Barto, 2018, Lagoudakis and Parr, 2003], include least squares temporal difference Q-learning (LSTD-Q) and least squares policy iteration (LSPI).

Ernst et al. work with non-linear (parametric or non-parametric) representation for Q-function [Ernst et al., 2005]. They introduce the ‘Fitted Q-Iteration’ method to fit the batch of samples to the Bellman equation using tree-based supervised learning methods. Castelletti et al. extend the approach to solve for multi-objective Markov decision processes in batch mode [Castelletti et al., 2012]. The neural fitted Q-iteration (NFQ) [Riedmiller, 2005] is a popular algorithm which models the Q-network with multi-layer perceptrons and fits the Bellman optimality equation over a given batch of trajectories $\mathcal{D}$. The error function is given by the Bellman residual

[Baird, 1995] defined as

$$e = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} (r_t + \gamma \max_{a \in A} Q_{\theta}(s_{t+1}, a) - Q_{\theta}(s_t, a_t))^2 \quad (3.9)$$

Experience replay is a similar concept which initializes a fixed capacity of a buffer to store a history of samples [Lin, 1992]. Deep Q-network also includes experience replay in its formalism and may naturally be used as a batch reinforcement learning algorithm by pushing pre-collected data into the buffer [Mnih et al., 2015]. The error equation is updated by replacing the target Q_{θ} network with another network $\hat{Q}_{\theta}$

$$e = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} (r_t + \gamma \max_{a \in A} \hat{Q}_{\theta'}(s_{t+1}, a) - Q_{\theta}(s_t, a_t))^2 \quad (3.10)$$

Kalashnikov et al. describe a Q-learning algorithm called QT-Opt that is able to learn effective vision-based robotic grasping strategies from logged grasping trials [Kalashnikov et al., 2018]. The algorithm uses stochastic optimization over the critic to select continuous actions and target values.

The DP methods, however, face the problem of distribution shift during testing, as well as training (Section 3.5.2). It is crucial to address the out-of-distribution action problem in order to effectively implement offline reinforcement learning using dynamic programming. Previous work has shown that restricting the Q-value evaluation only to actions in the dataset avoids erroneous Q-values due to extrapolation error [Fujimoto et al., 2019]. A number of more recent works also observe that a variety of constraints can be used to mitigate action distribution shift [Wu et al., 2019].

3.6 Off-Policy Policy Evaluation Methods

Off-policy policy evaluation (OPE) is a class of methods to estimate the value of a trained policy using trajectories collected from one or more independent behavior policies [Sutton and Barto, 2018]. These techniques play a pivotal role in our work

as methods to compare policies trained with different algorithms and the tuning of the hyper-parameters. Similarly, there are numerous other real-world applications where an offline assessment of a policy is crucial before deploying it in the real environment. The early works on OPE relied on classical importance sampling to handle the distribution mismatch between the behavior policy(ies) and the evaluation policy [Precup, 2000]. In the past couple of years, techniques are being developed which estimate the Q-model of the policy and possibly combine it with the importance sampling methods. Voloshin et al. [Voloshin et al., 2019] categorize the OPE techniques as Inverse Propensity Score (IPS), Direct Methods (DM) and Hybrid Methods (HM).

Inverse Propensity Scoring (IPS)

The key idea behind IPS is to reweigh the rewards by the importance sampling (IS) weight [Precup, 2000]. Importance sampling weight for the τ_i trajectory of horizon T in the existing historical data is defined between the evaluation policy π_e and the behavior policy π_b as $\rho_{j:j'}^i = \rho_{j:j'}(\tau^i, \pi_e, \pi_b) = \prod_{t=j}^{\min(T-1, j')} \frac{\pi_e(a_t^i | s_t^i)}{\pi_b(a_t^i | s_t^i)}$. The value of the evaluation policy under IS is then calculated as

$$\hat{V}(\pi_e) = \sum_{i=1}^N \frac{\rho_{0:T-1}^i}{N} \sum_{t=0}^{T-1} \gamma^t r_t \quad (3.11)$$

Some of the variants of IS include weighted importance sampling (WIS) and step-wise WIS. WIS replaces the normalization factor N by $w = \frac{1}{N} \sum_{i=1}^N \rho^i$ and the step-wise WIS reweighs the sample rewards at each time step of the trajectory. The value of the evaluation policy under step-wise importance sampling is given as

$$\hat{V}(\pi_e) = \sum_{i=1}^N \sum_{t=0}^{T-1} \gamma^t \frac{\rho_{0:t}^i}{w_{0:t}} r_t \quad (3.12)$$

Direct Methods (DM)

Direct Methods focus on estimating the value functions of the evaluation policy (V^{π_e} or Q^{π_e}). One approach is to estimate the transition probabilities and reward function from the historical data, and use this approximate model as the MDP to

learn about the new policy [Paduraru, 2013, Jiang and Li, 2015a]. In the alternate approach, if a parametrized action-value function $\hat{Q}(s, a; \theta)$ is learned then value of the policy may be estimated as

$$\hat{V}(\pi_e) = \frac{1}{N} \sum_{i=1}^N \sum_{a \in A} \pi_e(a|s_0^i) \hat{Q}(s_0^i, a; \theta) \quad (3.13)$$

Hybrid Methods (HM)

The hybrid methods combine the aspects of both direct methods and importance sampling methods. Doubly robust-like approaches fall under this category. In the classical unbiased doubly robust (DR) OPE [Jiang and Li, 2015b], direct method is used to lower the variance of estimates from the importance sampling techniques (Equation 3.14).

$$\hat{V}(\pi_e) = \frac{1}{N} \sum_{i=1}^N \hat{V}(s_0^i) + \sum_{i=1}^N \sum_{t=0}^{T-1} \gamma^t \rho_{0:t}^i (r_t^i - \hat{Q}(s_t^i, a_t^i) + \gamma \hat{V}(s_{t+1}^i)) \quad (3.14)$$

The weighted doubly robust (WDR) [Thomas and Brunskill, 2016] is an adaptation to DR which self-normalizes the importance weights (similar to WIS). Thomas et al. introduce ‘model and guided importance sampling combined’ (MAGIC) that uses adaptive switching between direct method and doubly robust [Thomas and Brunskill, 2016].

Chapter 4

MDP FORMULATION FOR ENGAGING HRI

In this thesis, we train a generation model of non-verbal behaviors, smiles and nods, as backchannels in a robot to engage humans during HRI. We formulate the problem as a Markov decision process (MDP) and use engagement to define rewards. Human state is a continuous space represented by using a history of speech features from the user’s audio and discrete action space is binary decision of generating a gesture. We use 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 learned from this batch data using novel batch-RL algorithms. A greedy policy can then implicitly be deduced from the learned Q-value function.

4.1 MDP Formulation

Markov decision processes characterize most RL architectures. It is built on the Markov Property and describes how a state reflects the history till that point and the conditional probability distribution of future states depends only upon the present state and the action taken at that time. In this work, we evaluate the performance of an agent when the user’s audio features are given as a continuous state representation. We desire the solution to this MDP to produce an optimum policy for backchannels that maximizes user engagement. Each component of the MDP is defined below.

4.1.1 State / Speech Features

The continuous state of the human is represented by a statistical summarization of speech features Mel-Frequency Cepstrum Coefficients (MFCCs) and prosody. 13-dimensional MFCC features are computed using 40 milliseconds sliding Hamming

window at intervals of 25 milliseconds and the 6-dimensional prosody comprises speech intensity, pitch, and confidence-to-pitch with their first derivatives. The concatenation of MFCCs and prosody features creates a 19-dimensional feature vector as in [Bozkurt et al., 2016] at a rate of 40Hz. From the past 1 second of history (40 samples), a feature summarization for the 19-dimensional features is performed (used in [Metallinou et al., 2013]), which computes 11 of the statistical quantities for each dimension. These quantities are mean, standard deviation, skewness, kurtosis, range, minimum, maximum, first quartile, third quartile, median and inter-quartile range. The resultant 209-dimensional *continuous* state feature describes the short-term distribution of the audio features.

4.1.2 Action / Backchannels

Agent’s action is a binary variable, indicating the absence or presence of the backchannel. The gestures of nods and smiles/laughs of the user with the behavioral policy in the dataset were labeled at a rate of 40 Hz.

4.1.3 Reward / Engagement

The reward is a scalar quantity that comes from the engagement measures of the user at calculated at a rate of 40 Hz. Our measure of engagement is based on the attentional involvement method proposed in [Rich et al., 2010], which is applicable to face-to-face collaborative HCI scenarios. Similar to the [Rich et al., 2010], 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 [Rich et al., 2010], 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 from 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 $T / (\text{no. of CEs})$

in T). As MTBCE is inversely proportional to engagement, similarly to [Rich et al., 2010], we use $\text{pace} = 1/\text{MTBCE}$ 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, a window size of 15 seconds was chosen after some analysis to avoid abrupt changes in rewards.

4.2 Off-Policy Batch Data

The abundance of human-to-human datasets makes it convenient to prepare them as a batch of trajectories for training with batch reinforcement learning. For this the dataset is pre-processed to create the tuples $\langle s_t, a_t, r_t, s_{t+1} \rangle$ for $t = 1 : T$. At time t , s_t is the state of the human defined from speech features, a_t is the backchannel action (smile/nod) by the agent and r_t is the reward defined in terms of engagement levels generated from the environment. We work with the IEMOCAP dataset [Busso et al., 2008], 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 dialogues 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.

In terms of the RL scheme, we assume that of the two actors one assumes the role of an agent who executes some behavior policy (i.e., generates actions in the form of backchannels). The agent interacts with an environment that is represented by the second actor who generates states and rewards. Figure 4.1 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. The state is defined from a history of audio features and reward also comes from a window size of 15 seconds since engagement is defined over large time windows. The action, however, is defined as an instantaneous event and is sampled at 40 Hz in this work.

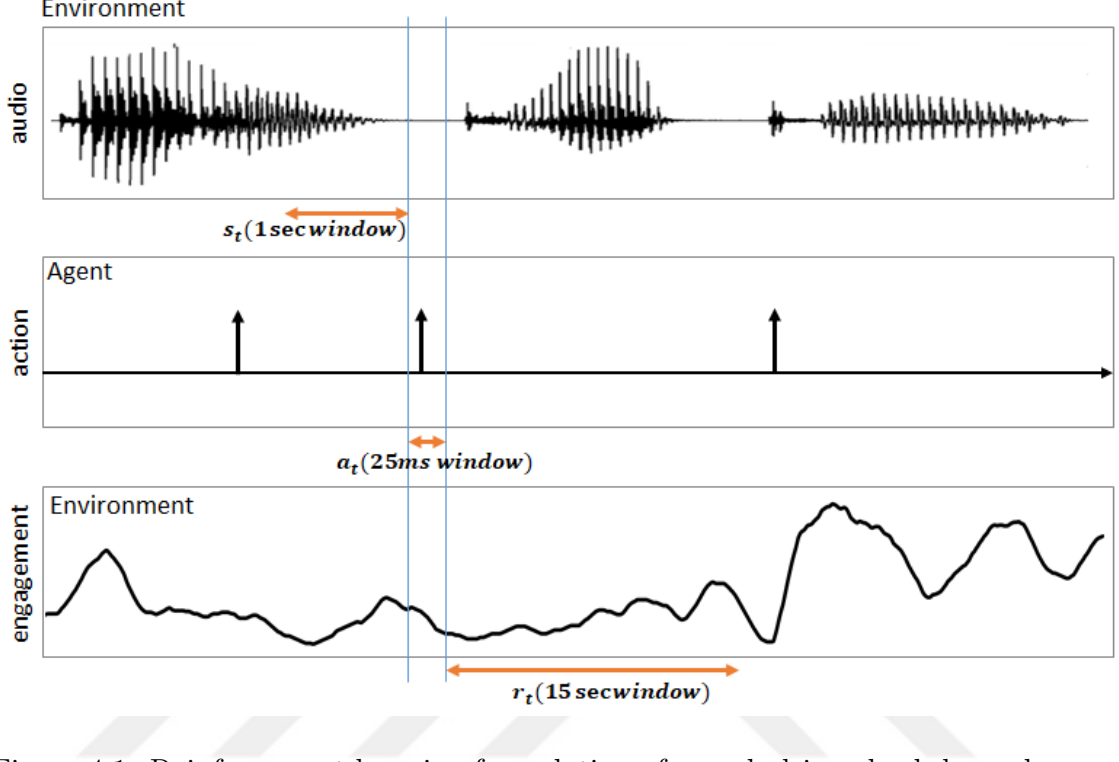


Figure 4.1: Reinforcement learning formulation of speech driven backchannel generation (not drawn to scale)

4.3 Offline Evaluation Methods

Prior to deploying a policy in a human-robot interaction experiment, it is first desirable to understand the policy's effectiveness using offline evaluation metrics. The following set of evaluation metrics are determined over the states present in the dataset: Bellman residual, the fraction of states where the policy triggers the gesture event, value of the policy using off-policy policy evaluation (OPE) techniques and statistical analysis of the time duration of the gesture event generated by the policy.

4.3.1 Bellman Residual

The Bellman residual given by equation 3.10, is computed over the entire batch at every epoch of the training. A smaller residual error means that the learned policy

is closer to the optimal policy and represents a true Q-function since it follows the Bellman equation more closely.

4.3.2 Off-Policy Policy Evaluation

After training, the values of the policies resulting from each training are evaluated with the step-wise weighted importance sampling estimator (WIS) given in equation 3.12. The values returned by WIS represent the expected cumulative discounted rewards estimated for each method.

4.3.3 Similarity to Human Behavior

The final metrics of interest are the similarity of the generated backchannel events to that initiated by a human. One way is to observe the frequency of occurrence of the backchannel by the new policy for the states available in the dataset. In the dataset 1.3% of the states were followed by a laugh/smile action and 2.5% of states resulted in a nod action.

Another way to analyze the backchannel event metric we take into consideration is the similarity of the duration of laughs that are generated by the robot to that of a human. For example, a policy that results in laughs which lasts several minutes is unnatural and will 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 a laugh here. In order to assess the naturalness of the laughs generated for the agent, we analyze the distribution of human laugh 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 min, max, mean and standard deviation.

Chapter 5

PROPOSED ALGORITHMS

5.1 Introduction

The extraction of transitions from the human-human interaction dataset was explained in Chapter 4, in order to create an off-policy batch of trajectories. The IEMOCAP dataset used in our work is not designed to study behaviors that engage the interacting partner and composes of a range of human behaviors that may engage or disengage. Therefore, learning to mimic humans in the dataset with supervised learning is not a viable approach. Instead, the learning algorithm should be able to recognize the samples that result in high rewards and build the policy that stays close to those regions in the state-action space. During the course of the research, we develop three approaches for training an engagement-boosting backchannel model in robots: (1) batch-DQN, (2) deep recurrent Q-learning (DRQL), and (3) sequential random deep Q-network (SRDQN). Batch-DQN gives an advantage with large datasets but loses sequential information by randomizing over the samples. On the other hand, DRQL incorporates sequential information into the learning process to handle the partial observability of the problem but is computationally expensive and hard to stabilize. The SRDQN algorithm is a more generalized version of the first two approaches. In this chapter, we first give details of each algorithm and present the respective results after training. In the end, we discuss and compare the advantages and disadvantages of each algorithm.

5.2 Approach I: Batch-DQN for Policy Optimization in MDP

Batch-DQN is a model-free off-policy algorithm for learning the robot's social behaviors in HRI in an offline manner. It uses only previously collected transition

samples $\langle s_t, a_t, r_t, s_{t+1} \rangle$ stored in an experience replay buffer $\mathcal{D}$ to learn the policy. The assumption is made that the state is fully-observable, and the interaction can be modeled with an MDP. Batch-DQN is particularly advantageous when the training data is large and the learning process may be time inefficient. It describes sample-selection criteria to choose only those samples that have a higher probability of being seen by the optimal policy. The knowledge from low-rewarding states is acquired in the early iterations and in the later stages, the policy is refined with high-rewarding samples.

5.2.1 Algorithm

Batch-DQN learns the optimal Q-value function $Q^*(s, a)$ from transition samples without any further interaction with the environment. Therefore, it falls into the class of offline Q-learning algorithms, which target to fit the Bellman control equation to the Q-network. Batch-DQN represents the function-approximation network Q_θ as multi-layer perceptron, with weights θ . The Bellman control equation is defined with a target network $Q_{\theta'}$ with weights θ' , similar to the DQN algorithm, and is given by the equation:

$$Q(s_t, a_t; \theta) = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a; \theta') \quad (5.1)$$

The standard assumption that the future rewards are discounted by a factor of $\gamma = 0.99$ per time-step is used here. Given target value $y_t = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a; \theta')$ and error $e = Q(s_t, a_t; \theta) - y_t$, defined as the difference between the right hand side and left hand side of Equation 5.1, the loss function $\mathcal{L}$ for a minibatch $\mathcal{B}$ is defined by the mean square error (MSE) loss function given by:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} e^2. \quad (5.2)$$

Batch-DQN introduces two key ideas into its framework. The first key idea in batch-DQN is that the capacity C of the experience replay buffer is initialized much smaller than the entire batch created from the dataset. While the IEMOCAP dataset has 12 hours of recordings, the replay buffer $\mathcal{D}$ stores 2 minutes of past

data. In each update step, samples from the next m time steps are pushed into $\mathcal{D}$ and an update is performed with a minibatch of m samples. Using replay buffer in this manner means the sampling distribution μ changes very little between gradient updates. This resembles online DQN where the sampling distribution changes gradually as new online samples are pushed into $\mathcal{D}$. On the other hand, NFQ randomizes samples over the entire batch. The actors in our human-human interaction dataset display a range of emotions, resulting in the underlying policies in different scenes to vary considerably. Randomly sampling from the entire batch will result in large differences in the updates. Gradient updates from slow varying sampling distribution using short-term replay buffers stabilizes the training.

The second key idea is to use a sample-selection criterion to push only a subset of samples into the replay buffer. The criterion is to either push a sample into the buffer $\mathcal{D}$ when the action a_t is consistent with the current greedy policy or to push with probability ϵ when a_t does not match the online policy. The initial value of ϵ is close to 1, so that in early iterations majority of the samples from the batch data are used in training. As training proceeds, ϵ decays exponentially, thus gradually converging to only those samples that correspond with the current agent policy. The detailed steps of the algorithm are given in Algorithm 1.

5.2.2 Offline Training

The transition tuples of the form $\langle s_t, a_t, r_t, s_{t+1} \rangle$ are prepared from the dataset, and are standardized to mean 0 and variance 1. We model the Q-function approximation network with a multi-layer perceptron of two hidden layer 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 MSE loss from the Bellman control equation is minimized using Adam optimizer. We have used a discount factor value 0.99 in all experiments. The detailed list of hyperparameters are given in Table 5.1.

Algorithm 1 Batch-DQN Algorithm (Refer to Table 5.1 for notations)

```

Initialize replay memory  $\mathcal{D}$  to capacity  $C$ 
Initialize  $Q_\theta$  with random weights  $\theta$ 
Initialize  $\hat{Q}_{\theta'}$  with weights  $\theta' = \theta$ 
Initialize  $\epsilon = \epsilon_0$ 
for epoch = 1 to  $T$  do
    while (transitions are available in the batch) do
        Get the minibatch of samples  $(s_t, a_t, r_t, s_{t+1})$  from next  $m$  time steps.
        for each sample in minibatch do
            if  $a_t == \operatorname{argmax}_a Q(s_t, a)$  then
                Push the sample into  $\mathcal{D}$ .
            else
                Push sample into  $\mathcal{D}$  with a probability  $\epsilon$ 
            end if
        end for
        Randomly select  $m$  samples from  $\mathcal{D}$ .
        Set  $y_t = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a; \theta')$ .
        Determine the MSE loss  $\mathcal{L}$  using Equation 5.2
        Perform a gradient descent step with respect to  $\theta$ .
        Every  $N$  steps reset  $\hat{Q} = Q$ .
    end while
    Exponentially decay  $\epsilon$  with rate  $\alpha$ 
end for

```

*5.2.3 Offline Evaluation**Bellman Residual*

To illustrate the effectiveness of batch-DQN over NFQ, Figure 5.1 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 5.2 gives the mean errors obtained over the

Table 5.1: Batch-DQN: List of hyper-parameters and their values.

Hyper-parameter	Symbol	Value	Description
minibatch size	m	512	Number of training examples over which gradient descent is performed
replay memory size	C	10000	Capacity of $\mathcal{D}$ from which a minibatch is sampled
history length	H	1 sec	States are defined from this much past audio data
target network update frequency	N	100	Frequency with which target network is updated (measured in number of parameter updates)
discount factor	γ	0.99	Discount factor used in Q-learning update
learning rate	l_r	0.00005	Initial learning rate used by the optimizer
initial exploration	ϵ_0	0.9	Initial value of ϵ
final exploration	ϵ_∞	0.05	Final value of ϵ in ϵ -greedy
exploration decay	α	10000	The decay rate of ϵ in terms of number of samples
optimizer	-	Adam	Optimization algorithm
activation function	-	ReLU	Non-linearity between each linear layer
weight initialization	-	normal random	Initialization method of neural network parameters

test sets.

Table 5.2: Batch-DQN: 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

Off-Policy Policy Evaluation

The off-policy policy evaluation results obtained from step-wise weighted importance sampling are shown in Table 5.3 over the entire dataset. 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

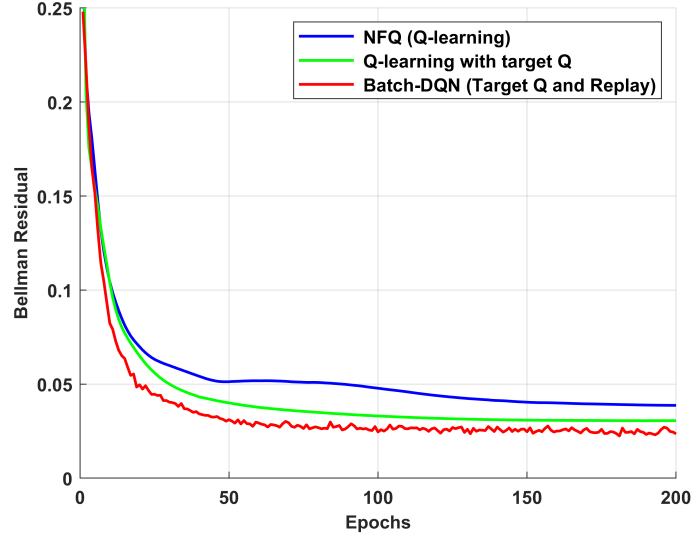


Figure 5.1: Bellman residual vs training epochs for batch-DQN and baseline NFQ algorithms

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 5.3: Batch-DQN: 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

Similarity to Human Behavior

Table 5.4 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 (~ 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 5.4: Similarity of agent laughs to human laughs

	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

5.3 Approach II: Deep Recurrent Q-Learning for Policy Optimization in POMDP

A common issue in real-world problems is the unavailability of complete state information, and instead only partial observations with some distribution over the true state are emitted by the environment. In other words, the Markov property rarely holds with real world environments. Approaching these problems with partially observable MDPs allows the dynamics of the system to be captured more closely, since the formulation explicitly acknowledges the observations as partial glimpses of the

states. Estimating Q-values from observation using vanilla Q-learning like neural fitted Q-iteration [Riedmiller, 2005] may be arbitrarily bad since $Q_\theta(s, a) \neq Q_\theta(o, a)$.

This chapter presents the experiments where we approach the problem of robot backchanneling behavior as a partially observable MDP. The work by Hausknecht et al. [Hausknecht and Stone, 2015] demonstrated that adding recurrency to the problem allows the Q-network network to better estimate the underlying system state, narrowing the gap between $Q_\theta(o, a)$ and $Q_\theta(s, a)$. Similarly, we use LSTM layers to model dependence through time between the inputs. The hidden state in LSTMs contains information from points earlier in the sequence and can be used as as part of the next input. Our experiments show the advantage of adding recurrency in the learning over a vanilla Q-learning approach.

5.3.1 Algorithm

Algorithm 2 Deep recurrent Q-learning (Refer to Table 5.5 for notations)

```

Initialize  $Q_\theta$  with random weights  $\theta$ 
Initialize  $\hat{Q}_{\theta'}$  with weights  $\theta' = \theta$ 
Initialize  $\alpha = \alpha_0$ 
for epoch = 1 to  $T$  do
    while (transitions are available in the batch) do
        Zero-initialize the hidden state
        Sample randomly a sequence of length  $L$ 
        Feed forward the sequence and hidden state to Q-networks
        Determine the loss  $\mathcal{L}$  with Q-value estimates for each time-step using Equation 5.3
        Clip the gradients between  $[-\alpha, \alpha]$ 
        Perform a gradient descent step with respect to  $\theta$ .
        Exponentially decay  $\alpha$ 
        Every  $N$  steps reset  $\hat{Q} = Q$ .
    end while
end for

```

Deep recurrent Q-learning falls in the category of model-free Q-learning algorithms, where the Bellman control equation is fitted to the batch data following

Table 5.5: List of hyper-parameters and their values

Hyper-parameter	Symbol	Value	Description
sequence length	L	80	Length of sequence passed to LSTM (equivalent to 2 seconds of data)
audio history	H	1 sec	Duration of audio data used to define the states.
target network update frequency	N	100	Frequency with which target network is updated (measured in number of parameter updates)
discount factor	γ	0.99	Discount factor used in Q-learning update
learning rate	l_r	0.00005	Initial learning rate used by the optimizer
initial clamp	α_0	0.001	Initial value of the factor α for clamping gradients
clamp decay	β	300	The decay rate of α in terms of number of updates
optimizer	-	Adam	Optimization algorithm
activation function	-	ReLU	Non-linearity between each linear layer
weight initialization	-	normal random	Initialization method of neural network parameters

an iterative procedure. It represents the function-approximation network Q_θ as a *fully connected LSTM structure*, with weights θ . As described in earlier chapter, the target network $Q_{\theta'}$ with weights θ' , has network structure identical to Q_θ . The loss function is defined as the mean square error between the right hand side and left hand side of the Bellman control equation and is given as:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} (Q_\theta(s_t, a_t) - (r_t + \gamma \max_a Q_{\theta'}(s_{t+1}, a)))^2. \quad (5.3)$$

The algorithm works by randomly selecting a sequence of L transitions. The length L is selected to be 80 samples which is equivalent to 2 seconds of data. Varying values of L were tested and we found $L = 80$ to be stable while improving the results. The sequence is input to the Q-network along with a zero-initialized hidden state. To further stabilize the training, the gradients are clamped with the parameter α , that decays with iterations. The algorithm steps are given in Algorithm 2.

5.3.2 Offline Training

The Q-function is modelled with a fully connected LSTM network (FC-LSTM), structured with 209 input layer size (state feature size), one hidden LSTM layer with 100 neurons, one hidden linear layer with 25 neurons and 2 outputs (Q-values for the two actions). ReLU is used as the nonlinear activation function between each layer. The MSE loss from the Bellman control equation is minimized using Adam optimizer. We have used a discount factor value 0.99 in all experiments. The detailed list of hyperparameters are given in Table 5.5.

For all our experiments the batch data is split into train and test sets in the ratio 4:1, hence 5-fold training is performed. It is split as leave one subject out (LOSO) and hence the reported results are subject independent. We test the performance of deep recurrent Q-learning with the baseline NFQ algorithm. For the NFQ method, identical Q-network is used, except the first LSTM layer which is replaced with a linear layer of the same size to form a multi-layer perceptron (MLP). We would also mention here that representation of Q-network with linear function approximator was found to be unstable. Furthermore, to compare with policy learned with supervised learning, we train another MLP network with action-labels from the dataset, without any notion of rewards. A softmax layer is appended to the network to produce a stochastic policy and the loss is defined by the cross-entropy loss function.

5.3.3 Offline Evaluation

Bellman Residual

Fig. 5.2 shows the training curves for the multi-layer perceptron (MLP) and fully connected LSTM networks. Both models converge to a stable point. However, the lower Bellman residuals from the LSTM model show better optimality and validity of its value function approximation network. For comparison purpose, we also present the effect of different lengths of sequences used for LSTM training. Though truncation of sequences to L samples may result in information loss from history, longer sequences become harder to train with LSTMs. The training curves for a

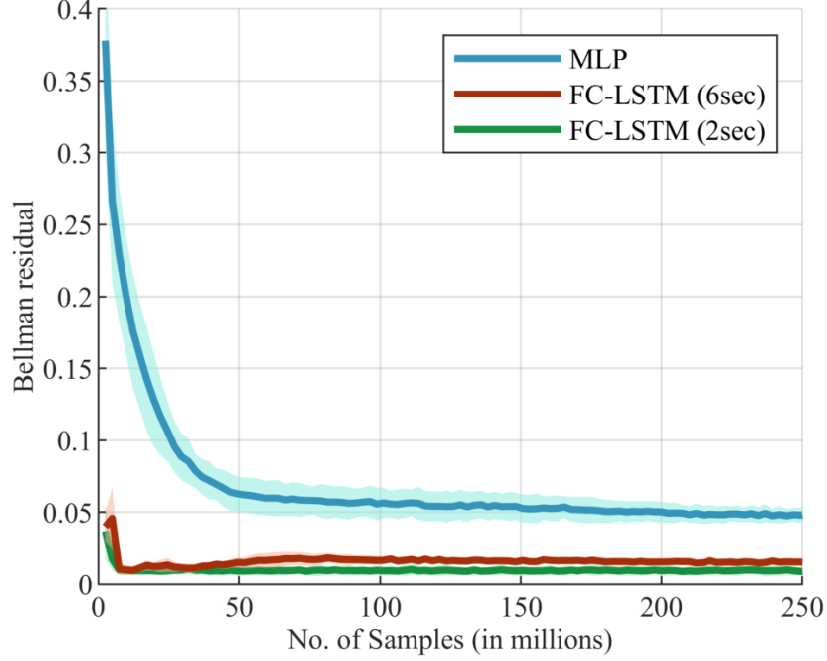


Figure 5.2: Bellman residual vs training samples for deep recurrent Q-learning with fully connected LSTM structure and the baseline NFQ with multi-layer perceptron architecture.

length of past 2 seconds of data versus past 6 seconds of data are presented here. Fig. 5.3 shows the trend of Q-values estimated by each network, averaged for the states present in the batch data. We observe that the estimates are close to each other, with LSTM network with 2 seconds of history surpassing marginally.

Off-Policy Policy Evaluation (OPE)

The training process for each technique produces an optimal Q-value function, which is used to implicitly define a policy. The simplest method is to act greedily and produce a deterministic policy by selecting the action with the highest Q-value. But since the OPE technique uses importance sampling and requires probabilities to reweigh each trajectory, we assign a high probability of 95% to the action suggested by the greedy policy. Also, it is observed that each agent produces laughs much

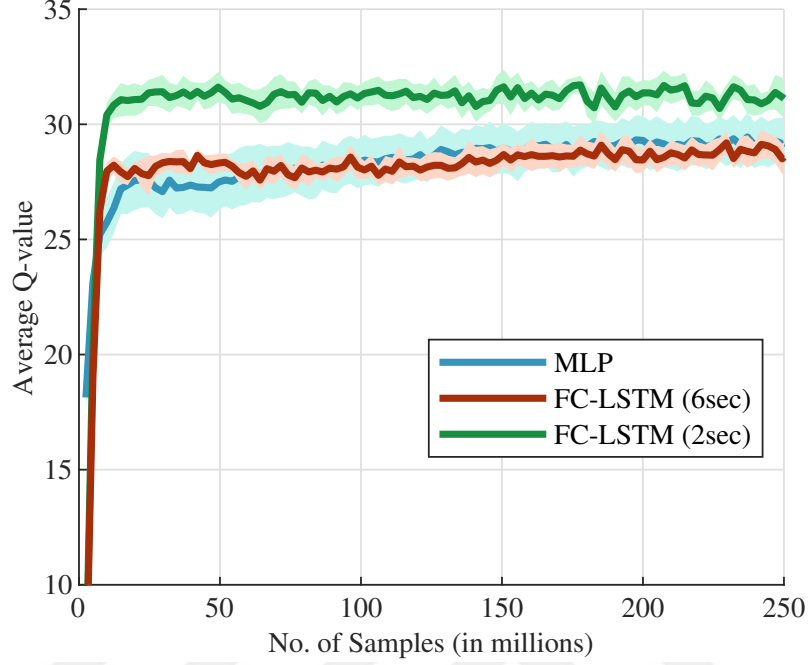


Figure 5.3: Estimated Q-value vs training samples for FC-LSTM Q-network and multi-layer perceptron Q-network

more frequently than the non-laughing event. On the contrary, in the dataset the laugh events occur only $\sim 1.5\%$. In order to control the number of laughs, we use the softmax function to convert Q-values to probabilities and apply different thresholds to define a new deterministic policy. The OPE is performed at different threshold values (or amount of generated laughs) and similar to before a 95% probability is assigned to the suggested action. Fig. 5.4 shows the values estimated by WIS at different fraction of laughs. The fully connected LSTM clearly outperforms MLP, producing a maximum value of 30.7 versus 24.8, whereas the batch data consists of an average value of 21.47. The policy from supervised learning is also tested in a similar fashion. Different thresholds are used to limit the number of laughs and OPE is performed from the deterministic actions suggested by the policy.

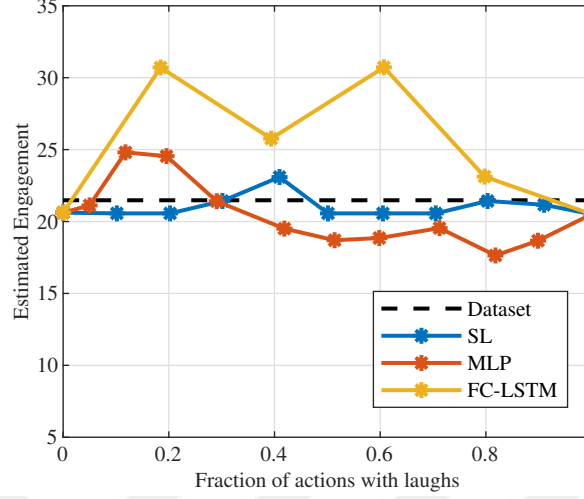


Figure 5.4: Value estimation with off-policy policy evaluation (OPE) for FC-LSTM and the baselines MLP and supervised learning (SL)

5.4 Approach III: Sequential Random DQN with Constraint for Policy Optimization in MDP

The work presented here is an extension of the preliminary version presented in Section 5.2 [Hussain et al., 2019b] and combines it with ideas from second algorithm in Section 5.3 [Hussain et al., 2019a]. The main contributions of the work in this section are as follows.

- The *Sequential-Random Deep Q-Network (SRDQN)* is proposed as a batch-RL algorithm to train an agent for non-verbal gestures. SRDQN introduces the key idea of using trajectories (of sequentially related samples) to make each update. This is contrary to batch-DQN where an update is made from random decorrelated samples. The results predicted more engaging interactions with the SRDQN training.
- We address the problem of distributional shift, where the distribution of states visited during training differ from the distribution of states visited during

testing, by using a constraint during the learning process. The *reward factor* (κ) is introduced as a hyper-parameter to control the backchannel predictions by the policy. It is tuned to bring the learned policy closer to the dataset policy, hence mitigating the distributional shift problem.

- We train our RL agent for two types of backchannels: nods and smiles. Both backchannels are trained independently with the SRDQN algorithm and their performances are compared. The work in Chapter 5.2 considered only smile generation. Important conclusions are drawn when the results of nod and smile policies are compared and contrasted.

5.4.1 Algorithm

We introduce the Sequential-Random Deep Q-Network (SRDQN) as an off-policy batch-RL algorithm for learning robot’s social behaviors in HRI. It belongs to the class of Q-learning algorithms where the model for rewards and transition dynamics are not explicitly learned. The quantity of interest is state-action value function, denoted $Q^\pi(s, a)$, which gives the expected future return starting from a particular state-action tuple. The goal of SRDQN is to learn the optimum Q-value function, $Q^*(s, a)$, using only previously collected transition samples $\langle s_t, a_t, r_t, s_{t+1} \rangle$ stored in an experience replay buffer $\mathcal{D}$.

SRDQN represents the Q-values with multi-layer perceptron function approximator, as proposed earlier by Riedmiller in neural fitted Q-iteration (NFQ) algorithm [Riedmiller, 2005]. It follows the iterative process of fitting the Bellman control equation to the samples in $\mathcal{D}$. The Q-values are estimated with neural network Q_θ , parameterized with weights θ . A separate network $\hat{Q}_{\theta'}$ determines target value, parameterized by θ' , that are periodically updated after every N target updates. The Bellman control equation takes a form similar to that proposed by deep Q-network (DQN) algorithm [Mnih et al., 2015], and is given by:

$$Q(s_t, a_t; \theta) = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a; \theta') \quad (5.4)$$

We make the standard assumption that the future rewards are discounted by a

Algorithm 3 Sequential-Random Deep Q-Network (Refer to Table 5.6 for notations)

```

Initialize replay memory  $\mathcal{D}$  to capacity  $C$ 
Initialize  $Q_\theta$  with random weights  $\theta$ 
Initialize  $\hat{Q}_{\theta'}$  with weights  $\theta' = \theta$ 
for epoch = 1 to  $T$  do
    while (transitions are available in the batch) do
        Get the trajectory of samples from next  $m$  time steps.
        Push the  $m$  samples to  $\mathcal{D}$ .
        Randomly choose a memory index  $i$  of  $\mathcal{D}$ .
        Select sequential  $m$  samples starting from index  $i$ .
        Set  $y_t = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a; \theta')$ .
        Determine the smooth L1 loss  $\mathcal{L}$  using Equation 5.5
        Perform a gradient descent step with respect to  $\theta$ .
        Every  $N$  steps reset  $\hat{Q} = Q$ .
    end while
end for

```

factor of $\gamma = 0.99$ per time-step. Given target value $y_t = r_t + \gamma \max_a \hat{Q}(s_{t+1}, a; \theta')$ and error $e = Q(s_t, a_t; \theta) - y_t$, the loss function is defined by the smooth L1 loss function as:

$$\mathcal{L} = \begin{cases} 0.5e^2, & \text{if } |e| < 1 \\ |e| - 0.5, & \text{otherwise} \end{cases} \quad (5.5)$$

Similar to batch-DQN algorithm (Chapter 5.2), the first key idea in SRDQN is maintaining a short-term experience replay buffer with capacity C . In our experiments, the capacity C is defined to store two minutes of past interaction data in the the replay buffer $\mathcal{D}$. The IEMOCAP dataset, on the other hand, has 12 hours of recordings. In each update step, samples from a trajectory of length m are pushed into $\mathcal{D}$ and an update is performed with some other randomly sampled trajectory from the buffer. As discussed earlier, updates with short-term buffers means the sampling distribution μ changes very little between consecutive target updates. While this resembles online DQN in some sense, it is a special case of NFQ

which randomizes samples over the entire batch. Target updates from slow varying sampling distribution using short-term replay buffers helps to stabilize the training.

The second key idea in SRDQN is the use of trajectory $\langle s_1, a_1, r_1, \dots, s_m, a_m, r_m, s_{m+1} \rangle$ of length m to make an update. This is contrary to DQN and batch-DQN where an update is made from random decorrelated samples. The replay buffer $\mathcal{D}$ in SRDQN stores samples sequentially. In each update step, a memory index i is randomly chosen from $\mathcal{D}$ (hence *random* of SRDQN). Starting from index i , m sequential samples are selected to create the minibatch (hence *sequential* of SRDQN). Thus, each update is performed from sequentially correlated samples. The problem of HRI is inherently a partially observable problem, where only observations (o) are available with some state distribution. Introducing sequentiality into the learning allows the Q-network to better estimate the underlying system state, narrowing the gap between $Q(o, a|\theta)$ and $Q(s, a|\theta)$, as previously discussed in Chapter 5.3. Algorithm 3 summarizes the learning steps.

5.4.2 Offline Training

A multi-layer perceptron (MLP) is used to model the function approximation network. Identical architectures are used for the Q-networks of nod and smile behaviors. The function approximator takes a 209-dimensional input vector (state feature size), has two hidden layers with 100 and 25 neurons respectively, and outputs two Q-values for the actions, ‘do nothing’ and ‘generate backchannel’ (nod or smile). ReLU activation function is applied between each linear layer. The Adam optimizer is used with an exponentially decaying learning rate, to update the network weights during training. We introduce the hyperparameter *reward factor* (κ) for scaling rewards during training. For $\kappa \leq 1$, the rewards of the batch data are modified such that:

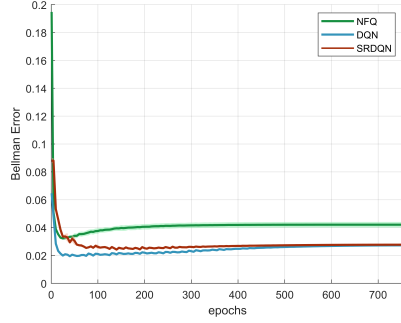
$$r'_t = \begin{cases} r_t, & \text{if } a_t = \text{“do nothing”} \\ \kappa r_t, & \text{if } a_t = \text{“generate backchannel”} \end{cases} \quad (5.6)$$

With $\kappa = 1$, the learned policy π is dominated with the action “generate backchannel” when it is evaluated with the states available in the dataset. On

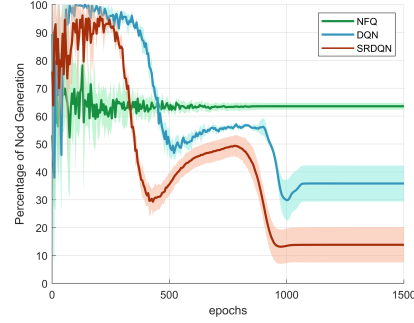
Table 5.6: List of hyperparameters and their values

Hyperparameter	Symbol	Value	Description
minibatch size	m	256	Number of training examples over which gradient descent is performed
replay memory size	C	4800	Capacity of $\mathcal{D}$ from which a minibatch is sampled
history length	H	1 sec	Duration of audio data used to define the states.
target network update frequency	N	100	Frequency with which target network is updated (measured in number of parameter updates)
discount factor	γ	0.99	Discount factor used in Q-learning update
learning rate	l_r	0.00005	Initial learning rate used by the optimizer
reward factor	κ	0.9	Factor for scaling rewards that are generated from backchannel actions
optimizer	-	Adam	Optimization algorithm
exponential LR scheduler	-	0.99	Exponential factor used to decay the learning rate per epoch
activation function	-	ReLU	Non-linearity between each linear layer
weight initialization	-	normal random	Initialization method of neural network parameters

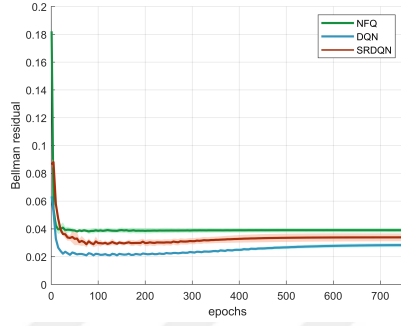
the other hand, the behavior policy π_b i.e. the human policy in the dataset scarcely generates a backchannel (2.5% of actions are nods and 1.3% are smiles). The large difference in the learned policy and the behavior policy indicates the distributional shift problem: while the function approximator is trained under one distribution, it might face a different distribution when interacting with the environment. The distributional shift is quite harmful since no guarantees can be made about errors on out-of-distribution inputs [Levine et al., 2020]. A common way to address this problem is to impose constraints on the learning process such that the learned policy comes closer to the behavior policy [Fujimoto et al., 2019, Agarwal et al., 2020]. In order to limit backchannel events from RL-policy, we scale down the rewards r_t in the $\langle s_t, a_t, r_t, s_{t+1} \rangle$ samples where the action a_t is “generate backchannel”, as given in Equation 5.6. In our experiment, κ is set to 0.9. The hyperparameters of the training process are given in Table 5.6.



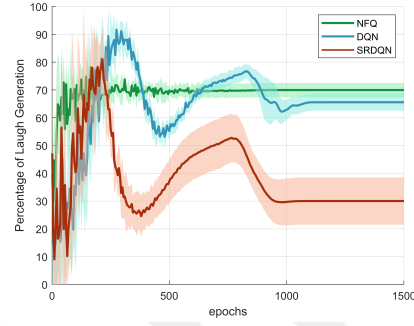
(a) Nods: Bellman Residual



(b) Nods: backchannel frequency



(c) Smiles: Bellman Residual



(d) Smiles: backchannel frequency

Figure 5.5: Bellman residual and backchannel frequency vs epochs for SRDQN and baselines NFQ and DQN algorithms.

Three batch reinforcement learning methods are implemented to train nod/smile behavior: neural fitted Q-iteration (NFQ) [Riedmiller, 2005], deep Q-network (DQN) [Mnih et al., 2015], and Sequential-Random Deep Q-Network (SRDQN). The dataset is partitioned in the ratio 4:1 for train and evaluation sets respectively. We perform the training five times for each method to address the variation seen in the converged policy when training is repeated. The greedy action from each Q-network is defined as $a = \operatorname{argmax}_a Q(s, a)$. The final backchannel policy is determined from the votes of each of the five greedy-policies. A majority vote of 3 or above triggers a backchannel.

5.4.3 Offline Evaluation

Bellman Residual

Figure 5.5 (a) and (c) shows the Bellman residuals for the test set plotted against the epoch number for nods and smiles respectively. The average value of the 5 training is represented by the solid line. Each curve shows a decreasing trend in the error, however, the final error for NFQ stabilizes at a value greater than that of DQN and SRDQN for both nods and smiles. The results of this first metric indicates an improved performance for DQN and SRDQN.

Backchannel Frequency

The NFQ policy predicts a high occurrence of nods (63%) and smiles (70%) at the end of the training. Figures 5.5 (b) and (d) show the ratio of event prediction as the training progresses and highlights the clear disadvantage of the NFQ algorithm. Although DQN improves the nod policy, it produces comparative results for smiles. The SRDQN clearly surpasses both algorithms and predicts the nods for 13% of the states and smiles for 30% of the states. These values are closer to the actual percentage of backchannels in the dataset (1.3% smiles and 2.5% nods). This is important for two reasons. First, since the dataset contain human-policy, the behavior policy of the batch is a near-optimal policy, and any learned policy is expected to be close to the behavior policy. Second, when the learned policy is close to the behavior policy, the distributional shift problem can be reduced.

Off-Policy Policy Evaluation

After training, the values of the policies resulting from NFQ, DQN and SRDQN were evaluated with the step-wise weighted importance sampling estimator (WIS) given in equation 3.12. 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. The stochastic policies required for the WIS technique is defined from the greedy policies such that 0.9 probability is assigned to the event

suggested by the greedy policy. Following the discussion of the work in [Raghu et al., 2018], the behavior policy π_b was estimated using approximate nearest neighbor [Hyvönen et al., 2016]. The off-policy policy evaluation results obtained from step-wise weighted importance sampling are shown in Table 5.7. It can be noted that the policy obtained by the SRDQN outperforms relative to the other two policies.

Table 5.7: Off-policy Policy Evaluation with step-wise Weighted Importance Sampling

	Estimated V^π			
	Dataset	NFQ	DQN	SRDQN
Nods	24.6	24.6	24.2	29.8
Laughs	24.6	26.5	20.0	27.8

Similarity to Human Behavior

Tables 5.8 and 5.9 shows the statistical similarity of backchannel event duration generated by each policy to that of a human. For the nods, even though the mean nod duration for the NFQ policy matches the closest with the human nods, it also recorded the largest difference in the standard deviation and maximum duration. The maximum length of nod was noted as 140 seconds, which is socially unacceptable by far. The SRDQN policy displayed the max and standard deviation most close to the human policy for nods. The minimum duration is fixed by our definition of the RL formulation. The table for smiles shows an even poorer result for NFQ and DQN policies, with very large chunks of sequential states observing a smile. SRDQN outperforms in all three measures of max, mean and standard deviation.

5.5 Discussion

In this chapter, three novel algorithms have been presented to address training of backchannel behavior models with batch-RL. The algorithms batch-DQN, DRQL

Table 5.8: Statistics of generated nodes (sec) in SRDQN vs baselines DQN, NFQ

	Min	Max	Mean	STD
Dataset	0.22	7.1	1.20	0.68
NFQ	0.10	140	1.50	2.81
DQN	0.10	21	0.84	1.20
SRDQN	0.10	6.3	0.33	0.32

Table 5.9: Statistics of generated smiles (sec) in SRDQN vs baselines DQN, NFQ

	Min	Max	Mean	STD
Dataset	0.08	5.6	0.92	0.85
NFQ	0.10	120	2.40	5.50
DQN	0.10	160	2.01	5.21
SRDQN	0.10	13	0.77	0.94

and SRDQN fall in the category of value-based Q-learning methods, where each method is characterized by their unique strengths and limitations. Batch-DQN has the advantage when large datasets are used for training, where convergence may be time consuming. It uses a selection-criterion to choose the samples which contain more information about the optimal policy. The less-rewarding samples, which have a low probability of being observed by the optimal policy, are gradually discarded. This reduces the computational burden and achieves faster convergence. However, batch-DQN is based on the assumption that the states are fully observable. On the contrary, real-world problems are generally partially observable, where only observations are available with some distribution over the states.

Deep recurrent Q learning (DRQL) acknowledges the partial-observability of our problem. The use of LSTM as the recurrent structure allows history of observations to be incorporated into the model in order to better infer the state information. Since LSTMs train with sequential data (i.e. trajectories), the sample-selection criterion of batch-DQN, for reducing the training data, cannot be applied to this method. The lower Bellman error reported with DRQL indicates its advantage. However, the problems of ‘exploding gradients’ and ‘vanishing gradients’ associated with LSTMs make the training challenging. Moreover, these recurrent structures generally require far more data, and the training is computationally expensive and

time-consuming.

The SRDQN algorithm targets to combine the advantages of batch-DQN and DRQL, that is, incorporates sequential information into the framework while avoiding complexity of the architecture. Moreover, batch-DQN and DRQL face the challenge of distributional shift. SRDQN with constraint mitigates this problem and learns a policy closer to the dataset policy, while improving performance over the baseline policies.



Chapter 6

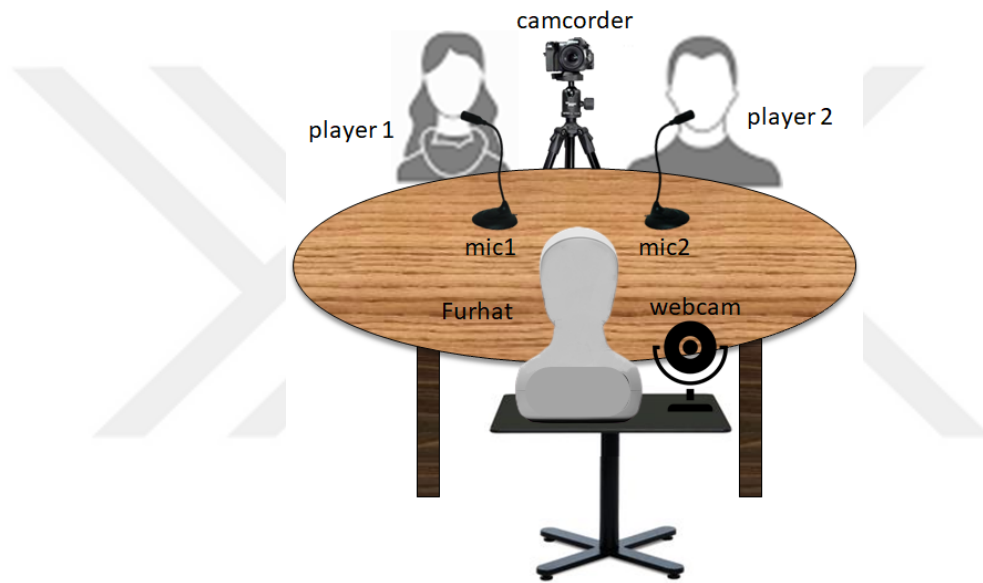
HUMAN-ROBOT INTERACTION EXPERIMENTS

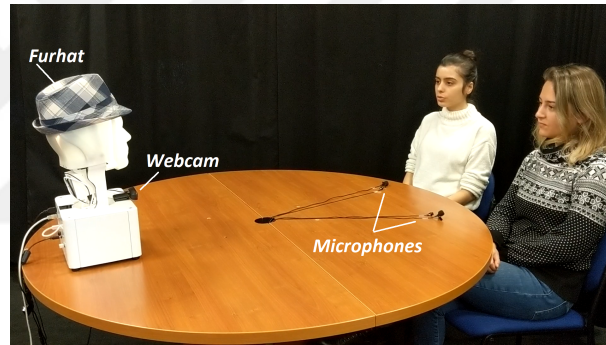
Figure 6.1: Human-robot interaction setup. Two participants sitting with Furhat around a circular table.

A human-robot interaction (HRI) experiment is designed to assess the effectiveness of trained backchannel in keeping the participants engaged. We target to evaluate the interaction by noting the engagement indicating signals triggered during the interaction (Section 2.3.3), and thorough a feedback questionnaire filled by the subjects. The experiment is designed with the back-projected robot head Furhat [Al Moubayed et al., 2013a, Al Moubayed et al., 2013b]. The hierarchical state machine implementation of Furhat facilitates the design of a conversation. In our experiments, the conversation is built on a story-shaping game, where a story is shaped according to the path selected by the player. A backchannel generation

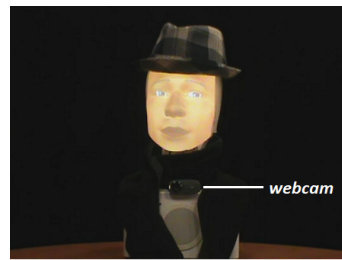
policy triggers smile and nod gestures through Furhat’s event handlers. Two models for backchannel behavior are implemented: baseline rule-based policy and test RL policy. The rule-based policy is defined such that the backchannels (nods and smiles) are generated at fixed time intervals. The test RL policy is selected to be the SRDQN trained policy, which shows the best performance with the offline evaluation metrics.

6.1 Experimental Design

6.1.1 Setup



(a) Experimental setup viewed from the side.



(b) Snapshot from the webcam. (c) Snapshot from the camcorder.

Figure 6.2: Snapshot of live human-robot interaction experiment with the robot Furhat and two participants

The experiment is designed as a two-person game with the robot leading the interaction. Figure 6.1 shows the arrangement for the experiment. In this setup, the two participants sit on fixed chairs next to each other and face Furhat around a

circular table. A webcam is set in front of the participants to capture their video and a camcorder is placed behind the participants to record the robot’s behavior. Two separate microphones are used for audio recordings of each participant. Snapshots of the experiment with the two participants are shown in Figure 6.2.

6.1.2 Procedure

A total of 24 participants take part in this user study, as pairs forming 12 groups. Among these 24 participants, there are 12 females and 12 males, with ages ranging from 20 years to 40 years and mean age 27 years. Each group is called to play the game twice consecutively on the same day, where each game differs in the generation policy of backchannels. The baseline game, with rule-based policy (game-RB), alternates smile and nod backchannels every two seconds. On the other hand, the test game with RL agent (game-RL) executes two policies independently: nod RL-policy and smile RL-policy. Each policy is learned with SRDQN and consists of five Q-networks. As explained in Section 5.4.2, the final decision for backchannel is made using the majority vote from the greedy decisions of the Q-networks. Half of the groups receive the game-RB as their first game and the other half receive the game-RL first.

Initially, the participants are given a briefing about the experiment and then left alone in the room to run through the game. At the beginning of the interaction, Furhat introduces itself and displays some of its capabilities such as winking and nodding. It then asks the question ‘Do you think I am a well-designed robot? What do you think of me?’. This initial interaction is not part of the experiment and targets to make the participants more comfortable and prepares them for the experiment. After the initial warm-up, the story-shaping experiment starts. A video recording of the experiment is saved for post-experiment feedback. After the experiments, the participants are asked to see the recordings and provide their feedback by filling a questionnaire. The questions are shown in Table 6.1 were asked and the participants rated these on a Likert scale of 5 points, with 1 meaning strongly

disagree and 5 strongly agree. Questions with similar contexts can be found in the user study paper by Sidner [Sidner et al., 2005]. These questions target measuring the effectiveness of the policy and the reliability of the experiment.

Table 6.1: Post-experiment questionnaire in the two categories: effectiveness of the policy and reliability of the experiment.

Effectiveness of the policy	
Q1	The interaction was engaging.
Q2	The robot's nods were timed appropriately.
Q3	The robot's smiles were timed appropriately.
Q4	The nods and smiles increased naturalness of the interaction.
Q5	The nods and smiles increased my interest in conversing with the robot.
Reliability of the experiment	
Q6	I knew what I was doing during the game.
Q7	I understood the robot well.
Q8	The robot understood me well.
Q9	The interaction went smooth

6.1.3 Story shaping game

The story-shaping game is a survival adventure sketched on a desert island where the captain and a sailor are stuck after a shipwreck. The plot of the game is designed using Furhat's dialog flow structure as a binary tree, with each node represents a scene from the story with two possible paths. Concealed behind a partition, a trained staff controls the story transition based on the participants' responses using a Wizard of Oz interface. To enrich the interaction, along with decision questions Furhat also asks discussion questions such as the rationale behind their decisions and how they feel. In the instances when Furhat is in listening mode, the backchannel policy takes the decision of generating a nod or smile. The game tree is 5 levels deep, with half of the leaves resulting in surviving the island. An example dialogue between the participants and Furhat is shown in Table 6.2.

Table 6.2: Example conversation between Furhat and two participants.

Furhat	For the past two months all of us have been travelling in an exploration ship along the Caribbean Sea. Unfortunately, we were caught in a storm and our ship hurled out of control. Now we find ourselves on a desert island. Captain, how do you find your surroundings?
Captain	Umm. It is mostly sand. And we are near the sea. At the beach. Just some trees. It's not a big island.
Furhat	Hmm. I see. [Looks at Sailor] The rescue forces have signaled they will reach us in three days. All we need to do is find some food. Now, should we explore around the beach or should we rather go and discover the forest.
Sailor	We should go the forest. It is better.
Furhat	Somebody seems like a nature lover. [Looks at Captain] Captain, what do you say?
Captain	I agree with Sailor. Let's discover the forest.
Furhat	It might be pretty scary in the forest. Why did you choose to go there?
Captain	Well we don't know if it is scary. And it might have some food and fresh water.
Furhat	Looks like you both agree. [Looks at Sailor]. Now we are in the forest. Sailor, what do you hear and see?
Sailor	Well I hear a lot of birds chirping and there are many trees. Bushes. And some squirrels are seen on the trees. Oh hey. There is a monkey.
Furhat	Ok. Guys, I hear some animal noises that may lead to meat. [Looks at Captain] But look in the other direction I see some fruit trees across a river. What do you think? Shall we go towards the river or hunt for animals?
Captain	Let's go towards the river. We don't have the tools to kill or capture animals.
Furhat	You do look very scared. [Looks at Sailor] Sailor let's hear your options.
Sailor	I will follow my Captain. Let's go to the river.
Furhat	Safe option. So hunting did not appeal you. Why?
Sailor	As my captain mentioned we don't have any tools to capture the animals.
Furhat	Since you both agree, no point in me having an opinion about it. We have reached the river.
	...

6.1.4 Architecture

The system architecture consists of three components communicating with each other, as shown in Figure 6.3. These are described below.

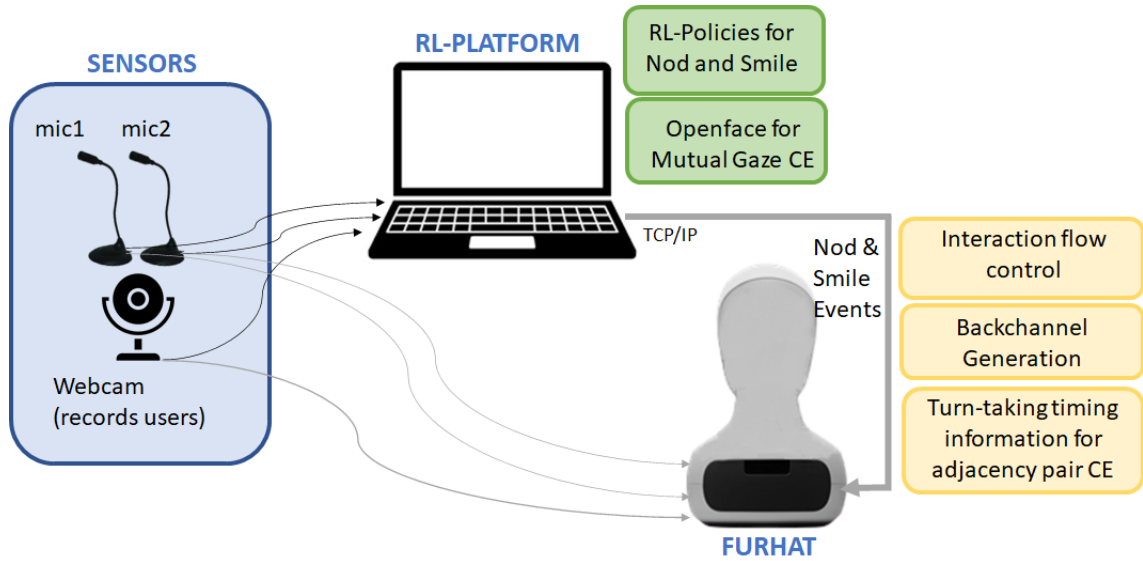


Figure 6.3: System architecture with three main components communicating with each other. (1) Audio and visual sensor signals are input to Furhat and RL-platform. (2) The RL-platform implements Q-networks for nods and smile, and concurrently runs the Openface application. (3) Furhat controls the conversation flow and triggers backchannels.

- The *audio-visual sensors* used in the experiment are two microphones and one webcam. The audio stream enters the RL-platform and Furhat system. The RL-platform processes the audio stream to extract audio features for state representation. Furhat receives the audio stream, along with the webcam's video stream, for its key functions such as multiple user tracking, head pose detection, automatic speech recognition etc. The audio streams from both participants and video streams from the webcam are logged for post-experiment analysis. Furthermore, the camcorder saves the recordings of Furhat's behaviors.
- The *RL-platform* is set up as a stand-alone c++ based system, which takes microphone audio as inputs and sends backchannel events over TCP/IP to Furhat. This platform implements the trained Q-networks for nod and smiles and handles the decision-making for generation of a backchannel. Furthermore,

the OpenFace software [Baltrusaitis et al., 2018] is used to determine the real-time gaze directions of each participant from the webcam video. A gaze event is triggered if the user exchanges gaze with Furhat within a time window when the robot switches its attention to the current user.

- *Furhat* conversational engine is structured as a state machine, with capabilities of speaking, listening and understanding. Its dialog flow structure facilitates the design of a story as a binary tree. At each listening turn of the robot, *Furhat*'s gestures/expressions unit triggered a backchannel if an action event was received from the RL=platform. The adjacency pair CE is determined online using speech events from the *Furhat* platform. An event is logged as successful if the user takes over his/her turn within a time interval after *Furhat* stops talking.

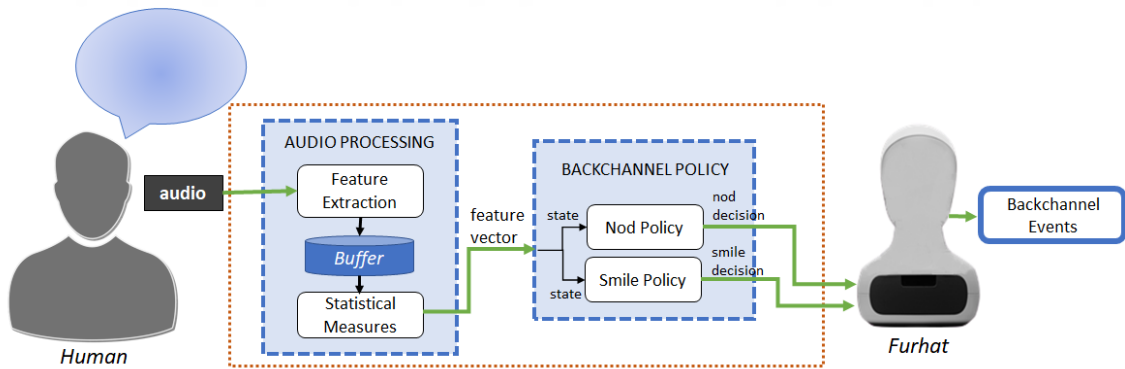


Figure 6.4: Backchannel policy implementation during robot's listening turn in the human-robot interaction experiment

The implementation of the RL backchannel policy during the experiment is shown in Figure 6.4. The input data stream is the audio captured from the microphone of the participant to whom *Furhat* is currently attending. The audio signal is first processed to extract audio features at a rate of 40 Hz (Chapter 4). A buffer stores the audio features from past one second of data. The final state vector

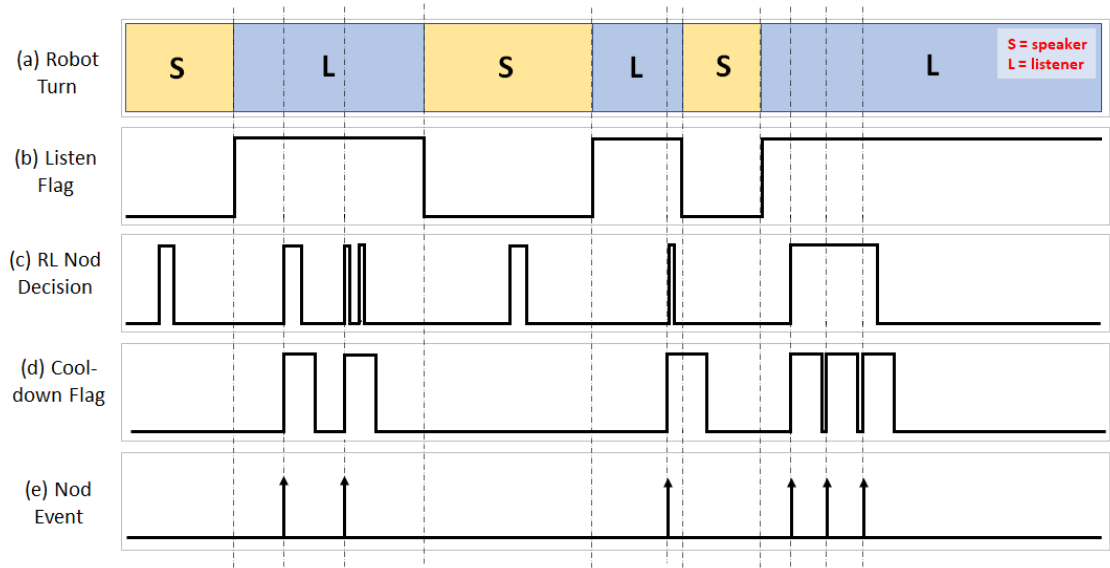


Figure 6.5: Backchannel event timing diagram for human-robot interaction experiment. (a) The robot’s listening turn and speaking turns are marked. (b) Backchannel flag is activated at every listening turn (c) The Q-network continuously takes decision irrespective of turn (d) The cool-down flag is activated for time 2 seconds in listening turns if a nod event is predicted. (e) Trigger points of nod event

is created from the statistical measures as described in Chapter 4. The frequency for the backchannel decision is selected to be 4Hz. At this rate, the nod and smile policies output an action decision during Furhat’s listening mode based on the state vector. To measure user engagement, the connection events ‘mutual gaze’, ‘adjacency pair’ and ‘backchannels’ are used (refer to Section 2.3.3). The backchannels (laughs, smiles and nods) are annotated offline after the experiment from the video recordings and cross-checked by a second annotator. The mutual gaze event in the RL-platform and the adjacency pair event in Furhat are recorded in real-time. User engagement values are calculated at 4 Hz after all the CEs are either logged online or annotated offline.

Figure 6.5 shows the timing diagram of the control flags used to govern the triggering of a backchannel. The listen flag is raised whenever Furhat enters listening mode. When this flag is inactive, all backchannel decisions by the Q-networks are

ignored. Such events represent false triggers since during Furhat’s speaking turn, its speech interferes with the user microphone signal. The *RL nod decision* is provided by the trained Q-network, irrespective of the robot’s turn. The length decisions of the nod event may also vary. The *cool-down flag* starts a timer of 4 seconds whenever a nod event is triggered. During this period, further backchannel decisions are inhibited. This flag controls the minimum time between two consecutive events. Finally, the *nod event* is triggered when the listening flag is raised, cool-down flag is deactivated and a nod decision is made by the RL-policy. Similar timing diagram can be sketched for the smile backchannel.

6.2 Experimental Results

The human-robot interaction sessions are evaluated subjectively as well as objectively. After the players complete the two games (game-RB & game-RL), they are asked to watch their own recordings and provide feedback with a questionnaire. This approach allows the subjects to interact with the robot more naturally during the experiment and focus better on the feedback questions later. The questions are shown in Table 6.1 and have similarity with those used by Sidner et al. in their user study [Sidner et al., 2005]. They are divided into two categories: *effectiveness of the policy* and *reliability of the experiment*. The participants score each question on a 5-point Likert scale ascending in the following order: strongly disagree, disagree, neutral, agree and strongly agree. The questions in the category ‘reliability of the experiment’ are used to discard the sessions which are unreliable due to failure of a component of the experiment. These problems may generate due to issues in the text to speech, speech recognition, state transitions or user understanding of the game. The results shown here are for the subjects who score the questions 6 to 9 with a rating greater or equal to 4. The questions in the category ‘effectiveness of the policy’ are designed to assess the effectiveness of the two policies in terms of engagement and naturalness.

For the objective analysis, we determine the engagement values of each par-

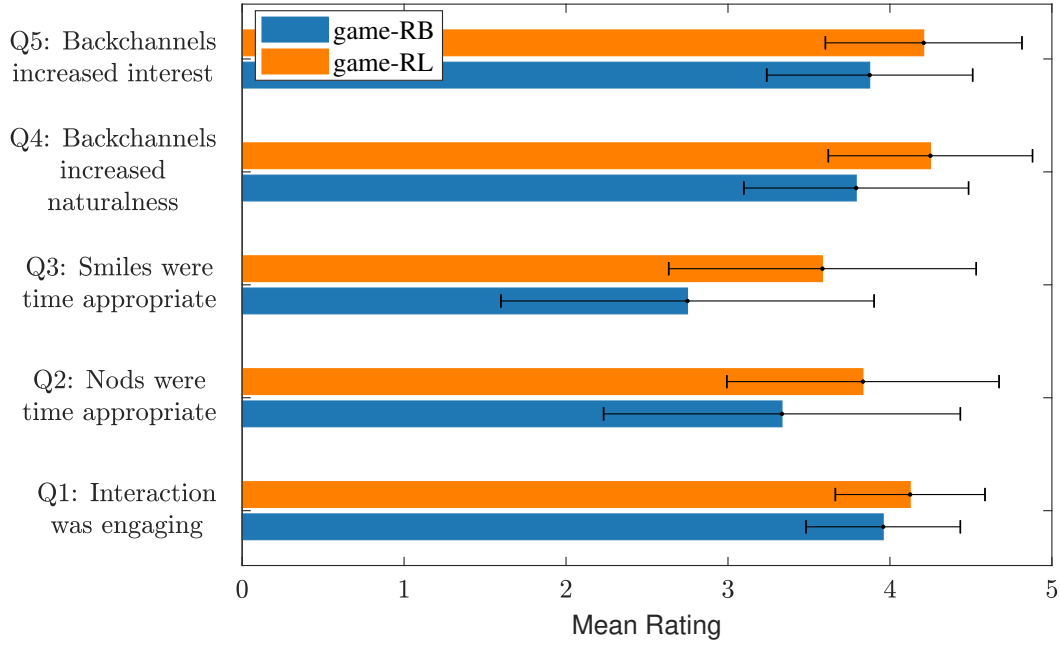


Figure 6.6: Mean user evaluation for the category ‘effectiveness of the policy’ in the post-experiment questionnaire for the game-RB and game-RL

participant using the annotated laughs, smiles, nods, and the online logged values of adjacency pair and mutual gaze events. The values are recorded at a rate of 4Hz. Therefore, with each interaction lasting about 5 minutes, there are on average 1200 engagement values per interaction for each player. Paired t-test analysis is performed to determine the significance of the results.

6.2.1 User-Feedback Analysis

The 24 participants provided their feedback by filling out the post-experiment questionnaires for game-RB and game-RL. Figure 6.6 shows the average ratings along with the standard deviations of each of the five questions in the category ‘effectiveness of the policy’. The higher rating of the game-RL not only confirms our claim about the effectiveness of the RL-agent, but some important conclusions may also be drawn from these values. Overall the participants agree that both games are en-

gaging, with scores of 3.88 and 4.08 for game-RB and game-RL respectively. A small difference is observed in the ratings between the two games. This is understandable since a complex number of technical and psychological factors may be involved in these ratings, such as the design of the game, smoothness of the interaction, mutual understanding between the players, robot behaviors and so on. In order to assess the effect of each backchannel, Q2 inquires about the time appropriateness of nods and Q3 asks the same for smiles. In Q2, game-RB scores 3.29 and game-RL scores 3.79 on average, while in Q3 game-RB was rated 2.71 and game-RL 3.58. An interesting observation is the greater tolerance of a simple repetitive policy for nods relative to a similar policy for smiles. In the rule-based agent, the smiles score a mean value below the ‘neutral’ response mark. This shows the higher complexity in the design of smile behaviors. Correspondingly, the ratings of the smiles generated by the RL agent improve by a greater margin. While Q1 inquires in general about engagement, Q4 and Q5 are designed to understand the role of backchannels in enhancing engagement. In these questions, game-RL receives better ratings than game-RB with a greater margin relative to Q1. The participants perceive the interaction in game-RL as more natural relative to game-RB (game-RB scored 3.71 and game-RL scored 4.21). Also, the higher score of 4.29 for game-RL versus 3.83 for game-RB in Q5 shows the players display greater interest and engagement when they interact with the RL-agent.

Furthermore, the statistical two-tailed paired t-test is used to determine whether the difference between the mean ratings of game-RB and game-RL is significant, where each group consists of 24 samples. The null hypothesis states that the values from the two groups come from the same distribution. The *p-values* for each of the five questions are shown in Table 6.3. Q4 and Q5 have $p < 0.05$ and hence support the significance of the RL agent policy in ensuring more naturalness and engagement during the interaction. While the results for smiles are significant (Q3), same is not concluded about nods. As earlier, a key finding from this analysis is that nods are less commonly perceived as ill-timed and may be accepted in a wider range

of scenarios. On the other hand, smiles need to be generated with more caution because context-less smiles result in a socially unacceptable behavior.

Table 6.3: The p-values of each question in the post-experiment questionnaire using the paired T-test

	Q1	Q2	Q3	Q4	Q5
p-value	0.2567	0.0761	0.0025	0.0183	0.0292

6.2.2 Engagement Analysis

Figure 6.7 shows the engagement values calculated for one participant during the interaction. Corresponding to the engagement values, the four connection events are shown below it.

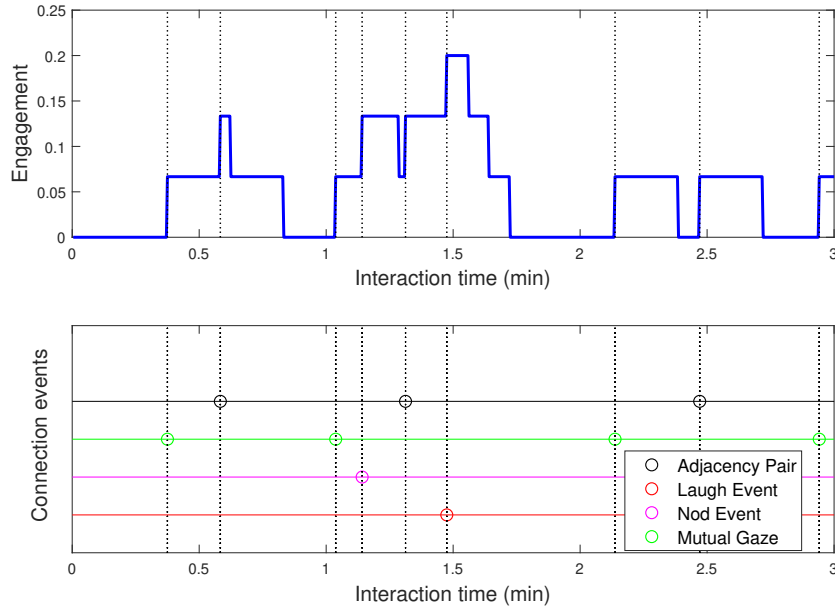


Figure 6.7: User engagement during the interaction for one participant (top row), and the corresponding detected/annotated connection events (bottom row)

The engagement values for the two types of games (game-RB and game-RL) are compared to get an objective assessment of the our method's performance. Figure

6.8 shows the cumulative engagement against time for the two types of games averaged over the interactions belonging to that group. The cumulative engagement at time t_i is the mean engagement over the history for $t \leq t_i$. The plot for initial 15 seconds is not shown since we define engagement over 15 second time intervals. The difference in the two curves shows the higher engagement of the participants when they play the game-RL. Another way to analyze the data is to first represent each interaction by a single engagement value. The mean value of the cumulative engagement is determined for each of the 48 interactions, where 24 samples belong to game-RB and 24 samples to the game-RL. The mean engagement for the game-RB is calculated as 0.0699 and for game-RL it is 0.0815. The paired t-test is performed for the statistical significance analysis and a p-value of 0.0361 ($p < 0.05$) indicates that the backchannels generated by RL-agent have a significant impact in improving the engagement of the users.

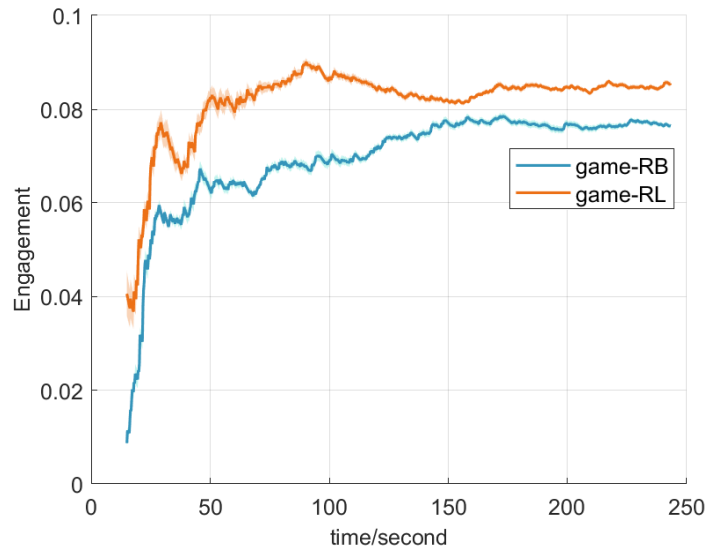


Figure 6.8: Cumulative engagement averaged over all participants

Chapter 7

CONCLUSION

7.1 *Summary of the Thesis*

The work presented in this thesis is divided into three parts: (I) motivation and literature review (II) problem formulation and offline training, and (III) user study with human-robot interaction experiment.

The first part highlights the impact of social robots in our daily lives and describes the plethora of applications offered by such robots. The thesis work is driven by the universal goal of user engagement during a human-robot interaction. To set the stage, a detailed literature review is given about the existing methods used to enhance user engagement. While a range of social behaviors of the robot impact user engagement, backchanneling as a feedback signal during listening turns is among the more influential factor. It includes verbal expressions, non-verbal vocalization and non-verbal gestures. We present a review of works that employ simple yet effective rule-based methods, primarily in tutoring and elderly care applications. A more modern approach for learning backchanneling in robots is by formulating the problem with reinforcement learning. After presenting the reader with basic concepts and elements of reinforcement learning, we discuss the rising success of RL in real-world problems. A research gap, however, exists in the design and implementation of offline reinforcement learning methods for training backchanneling models in robots. A detailed review of existing offline reinforcement learning techniques is provided. Furthermore, examples are provided from domains such as robotics, autonomous driving, language and dialog and healthcare, where offline RL techniques are already being applied successfully.

In the second part of the thesis, we present the mathematical formulation of our problem: *training backchanneling behavior models in a robot for enhancing user engagement in HRI*. We model the human-robot interaction as a Markov decision process (MDP) by exploiting user engagement as rewards. The MDP formulation is the core of the thesis, and forms the foundation for all experiments. In our MDP, the optimal policy takes decision for triggering backchannels conditioned a speech-based state representation, that maximizes user engagement. In order to employ offline reinforcement learning techniques, we work with human-human interaction dataset and extract transitions (s_t, a_t, r_t, s_{t+1}) from the dataset. A major issue with offline methods is the offline evaluation of the trained policy. *Bellman residual*, *off-policy policy evaluation* and *similarity to human behavior* are the offline metrics used in our work to assess the performance of the learned policies. We propose three value-based offline batch reinforcement learning algorithms to solve the problem. They fall into the category of Q-learning methods, where the optimal Q-network is learned by fitting the Bellman control equation. *Batch-DQN* uses the concept of short-term replay buffers and a sample-selection criteria to perform updates. *Deep recurrent Q-learning (DRQL)* targets to incorporate more history into the learning process and uses recurrent networks to model the Q-network. It solves for a partially-observable MDP, where we assume the true state information is unknown. *Sequential Random Deep Q-network (SRDQN)* incorporates sequential information into the learning with simpler architecture. A constraint is also employed to handle the short-comings of batch-DQN and DRQL like the distributional shift problem. Offline training with SRDQN, exceeded performance from all previous methods.

The last part of the thesis provides the experimental work performed in a human-robot interaction setting with an expressive 3d robot head. The interaction is designed as a story-shaping game, where the decisions from the players unfold the story. The game tree is implemented with Furhat’s hierarchical state machine platform. During the robot’s listening turns, the learned RL-policy is used to generate nod and smile backchannels by the robot to engage the users. We use the qualitative

as well as quantitative analysis methods to evaluate the performance of the policy. User feedback with a questionnaire provides insight into how the players perceive the backchanneling policy. Engagement indicators detected from the players are used to infer about engagement quantitatively. The higher acceptability of the RL policy versus a baseline policy is indicated by the statistically significant differences in the evaluation scores.

7.2 Discussion

The central theme of the thesis work is user engagement in a human-robot interaction. By exploiting engagement as rewards of a MDP, we target to obtain an optimum solution for a backchannel policy that maximizes engagement. The choice of features to define human state in HRI is not trivial. A complex set of audio-visual features seem a plausible choice for true state representation. However, it leads to the ‘curse of dimensionality’ problem of RL. We opt to utilize speech features for state representation. An important finding is the inadequacy of features from short windows, which result in unstable training. Feature summarization from a history of observation is found to be an effective representation of states. In order to preclude the tedious online training directly with humans, we design our problem for batch-RL methods. Our research is the first work that utilizes human-human interaction dataset to train a robot for backchannel behavior offline with batch reinforcement learning.

The three value-based batch-RL algorithms: batch-DQN, deep recurrent Q-learning (DRQL) and sequential random DQN (SRDQN), proposed during the course of research, out-perform the NFQ baseline. However each of them face individual limitations. While large collection of samples can allow the RL algorithm to learn a near-optimal policy, it also causes slow convergence. The sample selection-criterion of batch-DQN helps to select a subset of samples which contain more information about the optimal policy. Similar to any online learning method, where the robot is exposed to entire state-action space, the low-rewarding samples

are gradually discarded from the learning process. This reduces the computational burden and achieves faster convergence. For real-world applications, the important challenge is to address the partial-observability of the process. DRQL is designed to explicitly acknowledge this. It incorporates sequential information with recurrent structures to gather more information about the true state information. Though a promising solution for POMDPs, deep recurrent Q-learning faces computational complexity, stability problems and time-consuming training. We address the issues with the SRDQN algorithm, that is simple to train with low complexity, and incorporates sequential information to handle partial observability of the problem. The most important challenge we face is counterfactual queries, where the learning algorithm predicts high rewards for the actions not present in the dataset. Distributional shift is inferred from the large difference in the learned policy and the dataset policy. We demonstrate the use of constraint with the SRDQN algorithm to reduce the gap between the learned policy and the dataset policy.

An important observation during our experiments is the difference in the human perception of robot smiles and that of robot nods. The baseline in the user study is defined by a periodic triggering of backchannels. However, the smile rating of the baseline policy is much lower than the nod rating, in naturalness and engagement. This shows a greater tolerance for uniform distribution of nod events during the interaction, relative to smile events. With the trained RL policy, the smile rating boosted by a significant level. The performance of RL nod policy improved only marginally. Therefore, the success of our approach was more prominent in policy optimization for smiling as a backchannel.

7.3 Future Work

The offline reinforcement formulation for robot backchanneling, presented in this thesis, is a generic framework for any robot behavior. This research can be extended to train a variety of social behaviors such as non-verbal behaviors including gaze control, facial expressions and body gestures, and verbal backchannels like ‘hmms’

and ‘yeahs’. Furthermore, in this work we demonstrated the strength of audio-based state representation for policy decisions. The state definition may be enriched with more complex user signals such as facial expressions, emotions and verbal content of speech. The refined state information is expected to be closer to the true state of the user, and partial observability of the problem will reduce. However, the size of the solution space of a problem grows exponentially with each additional feature describing the state [Kaelbling et al., 1996]. Hence the challenge of “the curse of dimensionality” is inevitable. Bigger collections of offline trajectories, along with methods that identify the more significant regions of state space need to be employed to solve for the problem.

The algorithms introduced here fall in the category of offline Q-learning methods. Though the proposed methods produced policies with appreciable performances, much work can be extended in the algorithm design. Batch algorithms with the approaches of policy gradients and actor-critic method need to be explored. These methods result in a stochastic policy and are preferred when the Q-network becomes too complex. Another future direction is experimenting with techniques that mitigate the *distributional shift* of batch RL algorithms. Viable solutions that combine constraints and supervision with reinforcement learning need to be explored. A final motivation for future is from a bigger picture in the design of social robots. We may not restrict ourselves to the goal of engagement, but multiple design objectives can be targeted. Steinfeld et al. list the common metrics used in the design of human-robot interaction, such as persuasiveness, trust, and compliance [Steinfeld et al., 2006]. With accurate quantification of these metrics, they may be incorporated as the reward function of reinforcement learning formulation.

BIBLIOGRAPHY

- [Agarwal et al., 2020] Agarwal, R., Schuurmans, D., and Norouzi, M. (2020). An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*. Contributed Talk at NeurIPS 2019 DRL Workshop.
- [Al Moubayed et al., 2013a] Al Moubayed, S., Beskow, J., and Skantze, G. (2013a). The furhat social companion talking head. In *Interspeech 2013, 14th Annual Conference of the International Speech Communication Association, August 25-29, 2013, Lyon, France*, pages 747–749.
- [Al Moubayed et al., 2013b] Al Moubayed, S., Skantze, G., and Beskow, J. (2013b). The furhat back-projected humanoid head–lip reading, gaze and multi-party interaction. *International Journal of Humanoid Robotics*, 10(01):1350005.
- [Anzalone et al., 2015] Anzalone, S. M., Boucenna, S., Ivaldi, S., and Chetouani, M. (2015). Evaluating the engagement with social robots. *International Journal of Social Robotics*, 7(4):465–478.
- [Auer et al., 2009] Auer, P., Jaksch, T., and Ortner, R. (2009). Near-optimal regret bounds for reinforcement learning. In *Advances in neural information processing systems*, pages 89–96.
- [Baird, 1995] Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier.
- [Baltrusaitis et al., 2018] Baltrusaitis, T., Zadeh, A., Lim, Y. C., and Morency, L. (2018). Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE In-*

ternational Conference on Automatic Face Gesture Recognition (FG 2018), pages 59–66.

- [Belle et al., 2011] Belle, A., Hobson, R., and Najarian, K. (2011). A physiological signal processing system for optimal engagement and attention detection. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 555–561. IEEE.
- [Belpaeme et al., 2018] Belpaeme, T., Kennedy, J., Ramachandran, A., Scassellati, B., and Tanaka, F. (2018). Social robots for education: A review. *Science robotics*, 3(21).
- [Bickmore et al., 2010] Bickmore, T., Schulman, D., and Yin, L. (2010). Maintaining engagement in long-term interventions with relational agents. *Applied Artificial Intelligence*, 24(6):648–666.
- [Blankertz et al., 2010] Blankertz, B., Tangermann, M., Vidaurre, C., Fazli, S., Sanelli, C., Haufe, S., Maeder, C., Ramsey, L. E., Sturm, I., Curio, G., et al. (2010). The berlin brain–computer interface: non-medical uses of bci technology. *Frontiers in neuroscience*, 4:198.
- [Bohus and Horvitz, 2014] Bohus, D. and Horvitz, E. (2014). Managing human-robot engagement with forecasts and... um... hesitations. In *Proceedings of the 16th international conference on multimodal interaction*, pages 2–9.
- [Bojarski et al., 2016] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.
- [Bozkurt et al., 2016] Bozkurt, E., Erzin, E., and Yemez, Y. (2016). Multimodal analysis of speech and arm motion for prosody-driven synthesis of beat gestures. *Speech Communication*, 85:29–42.

- [Brewka, 1996] Brewka, G. (1996). Artificial intelligence—a modern approach by stuart russell and peter norvig, prentice hall. series in artificial intelligence, englewood cliffs, nj. *The Knowledge Engineering Review*, 11(1):78–79.
- [Broekens et al., 2009] Broekens, J., Heerink, M., Rosendal, H., et al. (2009). Assistive social robots in elderly care: a review. *Gerontechnology*, 8(2):94–103.
- [Brown et al., 2013] Brown, L., Kerwin, R., and Howard, A. M. (2013). Applying behavioral strategies for student engagement using a robotic educational agent. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4360–4365. IEEE.
- [Busso et al., 2008] Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335.
- [Cabi et al., 2019] Cabi, S., Colmenarejo, S. G., Novikov, A., Konyushkova, K., Reed, S., Jeong, R., Zolna, K., Aytar, Y., Budden, D., Vecerik, M., et al. (2019). A framework for data-driven robotics. *arXiv preprint arXiv:1909.12200*.
- [Cabibihan et al., 2013] Cabibihan, J.-J., Javed, H., Ang, M., and Aljunied, S. M. (2013). Why robots? a survey on the roles and benefits of social robots in the therapy of children with autism. *International journal of social robotics*, 5(4):593–618.
- [Campa, 2016] Campa, R. (2016). The rise of social robots: a review of the recent literature. *Journal of Evolution and Technology*, 26(1).
- [Castelletti et al., 2012] Castelletti, A., Pianosi, F., and Restelli, M. (2012). Tree-based fitted q-iteration for multi-objective markov decision problems. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

- [Chua et al., 2018] Chua, K., Calandra, R., McAllister, R., and Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765.
- [Codevilla et al., 2018] Codevilla, F., Miiller, M., López, A., Koltun, V., and Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE.
- [Darling, 2012] Darling, K. (2012). Extending legal rights to social robots. In *We Robot Conference, University of Miami*, volume 23.
- [Dasari et al., 2019] Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckepeper, K., Singh, S., Levine, S., and Finn, C. (2019). Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*.
- [Deisenroth and Rasmussen, 2011] Deisenroth, M. and Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472.
- [Doherty and Doherty, 2018] Doherty, K. and Doherty, G. (2018). Engagement in hci: conception, theory and measurement. *ACM Computing Surveys (CSUR)*, 51(5):1–39.
- [Ebert et al., 2018] Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., and Levine, S. (2018). Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*.
- [Ernst et al., 2005] Ernst, D., Geurts, P., and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556.

- [Fong et al., 2003] Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3-4):143–166.
- [Fujimoto et al., 2019] Fujimoto, S., Meger, D., and Precup, D. (2019). Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062.
- [Glas and Pelachaud, 2015] Glas, N. and Pelachaud, C. (2015). Definitions of engagement in human-agent interaction. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 944–949. IEEE.
- [Gordon et al., 2016] Gordon, G., Spaulding, S., Westlund, J. K., Lee, J. J., Plummer, L., Martinez, M., Das, M., and Breazeal, C. (2016). Affective personalization of a social robot tutor for children’s second language skills. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- [Guez et al., 2008] Guez, A., Vincent, R. D., Avoli, M., and Pineau, J. (2008). Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *AAAI*, pages 1671–1678.
- [Haarnoja et al., 2018] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.
- [Hall et al., 2005] Hall, L., Woods, S., Aylett, R., Newall, L., and Paiva, A. (2005). Achieving empathic engagement through affective interaction with synthetic characters. In *International Conference on Affective Computing and Intelligent Interaction*, pages 731–738. Springer.
- [Hausknecht and Stone, 2015] Hausknecht, M. and Stone, P. (2015). Deep recurrent q-learning for partially observable mdps. *CoRR*, abs/1507.06527, 7(1).

- [Heess et al., 2015] Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. (2015). Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pages 2944–2952.
- [Henderson et al., 2008] Henderson, J., Lemon, O., and Georgila, K. (2008). Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34(4):487–511.
- [Hester et al., 2018] Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. (2018). Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [Hussain et al., 2019a] Hussain, N., Erzin, E., Sezgin, T. M., and Yemez, Y. (2019a). Batch recurrent q-learning for backchannel generation towards engaging agents. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–7. IEEE.
- [Hussain et al., 2019b] Hussain, N., Erzin, E., Sezgin, T. M., and Yemez, Y. (2019b). Speech driven backchannel generation using deep q-network for enhancing engagement in human-robot interaction. *Proc. Interspeech 2019*, pages 4445–4449.
- [Hyvönen et al., 2016] Hyvönen, V., Pitkänen, T., Tasoulis, S., Jääsaari, E., Tuomainen, R., Wang, L., Corander, J., and Roos, T. (2016). Fast nearest neighbor search through sparse random projections and voting. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 881–888. IEEE.
- [Ishii and Nakano, 2008] Ishii, R. and Nakano, Y. I. (2008). Estimating user’s conversational engagement based on gaze behaviors. In *International Workshop on Intelligent Virtual Agents*, pages 200–207. Springer.

- [Ishii and Nakano, 2010] Ishii, R. and Nakano, Y. I. (2010). An empirical study of eye-gaze behaviors: Towards the estimation of conversational engagement in human-agent communication. In *Proceedings of the 2010 workshop on Eye gaze in intelligent human machine interaction*, pages 33–40.
- [Janner et al., 2019] Janner, M., Fu, J., Zhang, M., and Levine, S. (2019). When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pages 12519–12530.
- [Jaques et al., 2019] Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. (2019). Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.
- [Jiang and Li, 2015a] Jiang, N. and Li, L. (2015a). Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*.
- [Jiang and Li, 2015b] Jiang, N. and Li, L. (2015b). Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*.
- [Johnson et al., 2016] Johnson, A. E., Pollard, T. J., Shen, L., Li-Wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.
- [Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.
- [Kahn et al., 2018] Kahn, G., Villafior, A., Ding, B., Abbeel, P., and Levine, S. (2018). Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE.

- [Kaiser et al., 2019] Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. (2019). Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*.
- [Kakade and Langford, 2002] Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274.
- [Kakade, 2002] Kakade, S. M. (2002). A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538.
- [Kalashnikov et al., 2018] Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. (2018). Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673.
- [Kandasamy et al., 2017] Kandasamy, K., Bachrach, Y., Tomioka, R., Tarlow, D., and Carter, D. (2017). Batch policy gradient methods for improving neural conversation models. *arXiv preprint arXiv:1702.03334*.
- [Kendall et al., 2019] Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V.-D., Bewley, A., and Shah, A. (2019). Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254. IEEE.
- [Konda and Tsitsiklis, 2000] Konda, V. R. and Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014.
- [Lagoudakis and Parr, 2003] Lagoudakis, M. G. and Parr, R. (2003). Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149.

- [Lambertz, 2011] Lambertz, K. (2011). Back-channelling: The use of yeah and mm to portray engaged listenership. *Griffith Working Papers in Pragmatics and Intercultural Communication*, 4(1/2):11–18.
- [Lange et al., 2012] Lange, S., Gabel, T., and Riedmiller, M. (2012). Batch reinforcement learning. In *Reinforcement learning*, pages 45–73. Springer.
- [Lathuilière et al., 2019] Lathuilière, S., Massé, B., Mesejo, P., and Horaud, R. (2019). Neural network based reinforcement learning for audio–visual gaze control in human–robot interaction. *Pattern Recognition Letters*, 118:61–71.
- [Leite et al., 2014] Leite, I., Castellano, G., Pereira, A., Martinho, C., and Paiva, A. (2014). Empathic robots for long-term interaction. *International Journal of Social Robotics*, 6(3):329–341.
- [Leite et al., 2013] Leite, I., Martinho, C., and Paiva, A. (2013). Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308.
- [Leite et al., 2011] Leite, I., Pereira, A., Castellano, G., Mascarenhas, S., Martinho, C., and Paiva, A. (2011). Modelling empathy in social robotic companions. In *International conference on user modeling, adaptation, and personalization*, pages 135–147. Springer.
- [Levine et al., 2020] Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.
- [Levine et al., 2018a] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018a). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436.

- [Levine et al., 2018b] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018b). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436.
- [Lin, 1992] Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321.
- [Litman and Forbes-Riley, 2006] Litman, D. J. and Forbes-Riley, K. (2006). Recognizing student emotions and attitudes on the basis of utterances in spoken tutoring dialogues with both human and computer tutors. *Speech communication*, 48(5):559–590.
- [Liu et al., 2008] Liu, C., Conn, K., Sarkar, N., and Stone, W. (2008). Online affect detection and robot behavior adaptation for intervention of children with autism. *IEEE transactions on robotics*, 24(4):883–896.
- [Maddern et al., 2017] Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2017). 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15.
- [Metallinou et al., 2013] Metallinou, A., Katsamanis, A., and Narayanan, S. (2013). Tracking continuous emotional trends of participants during affective dyadic interactions using body language and speech information. *Image and Vision Computing*, 31(2):137–152.
- [Mitsunaga et al., 2006] Mitsunaga, N., Smith, C., Kanda, T., Ishiguro, H., and Hagita, N. (2006). Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. *Journal of the Robotics Society of Japan*, 24(7):820–829.
- [Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G.,

- et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.
- [Mo et al., 2018] Mo, K., Li, H., Lin, Z., and Lee, J.-Y. (2018). The adobeindoornav dataset: Towards deep reinforcement learning based real-world indoor robot visual navigation. *arXiv preprint arXiv:1802.08824*.
- [Monkaresi et al., 2016] Monkaresi, H., Bosch, N., Calvo, R. A., and D’Mello, S. K. (2016). Automated detection of engagement using video-based estimation of facial expressions and heart rate. *IEEE Transactions on Affective Computing*, 8(1):15–28.
- [Moro et al., 2019] Moro, C., Lin, S., Nejat, G., and Mihailidis, A. (2019). Social robots and seniors: A comparative study on the influence of dynamic social features on human–robot interaction. *International Journal of Social Robotics*, 11(1):5–24.
- [Mubin et al., 2018] Mubin, O., Ahmad, M. I., Kaur, S., Shi, W., and Khan, A. (2018). Social robots in public spaces: A meta-review. In *International Conference on Social Robotics*, pages 213–220. Springer.
- [Nagabandi et al., 2018] Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. (2018). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE.
- [Nie et al., 2019] Nie, X., Brunskill, E., and Wager, S. (2019). Learning when-to-treat policies. *arXiv preprint arXiv:1905.09751*.
- [O’Brien and Toms, 2008] O’Brien, H. L. and Toms, E. G. (2008). What is user engagement? a conceptual framework for defining user engagement with technology. *Journal of the American society for Information Science and Technology*, 59(6):938–955.

- [Osband et al., 2016] Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034.
- [O’Donoghue et al., 2018] O’Donoghue, B., Osband, I., Munos, R., and Mnih, V. (2018). The uncertainty bellman equation and exploration. In *International Conference on Machine Learning*, pages 3836–3845.
- [Paduraru, 2013] Paduraru, C. (2013). *Off-policy evaluation in Markov decision processes*. PhD thesis, McGill University Libraries.
- [Pan et al., 2017a] Pan, X., You, Y., Wang, Z., and Lu, C. (2017a). Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952*.
- [Pan et al., 2017b] Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., and Boots, B. (2017b). Agile autonomous driving using end-to-end deep imitation learning. *arXiv preprint arXiv:1709.07174*.
- [Parmas et al., 2018] Parmas, P., Rasmussen, C. E., Peters, J., and Doya, K. (2018). Pippis: Flexible model-based policy search robust to the curse of chaos. In *International Conference on Machine Learning*, pages 4065–4074.
- [Peters et al., 2009] Peters, C., Castellano, G., and De Freitas, S. (2009). An exploration of user engagement in hci. In *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots*, pages 1–3.
- [Peters et al., 2005] Peters, C., Pelachaud, C., Bevacqua, E., Mancini, M., Poggi, I., and Tre, U. R. (2005). Engagement capabilities for ecas. In *AAMAS’05 workshop Creating Bonds with ECAs*.
- [Pietquin et al., 2011] Pietquin, O., Geist, M., Chandramohan, S., and Frezza-Buet, H. (2011). Sample-efficient batch reinforcement learning for dialogue management

optimization. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):1–21.

[Pinto and Gupta, 2016] Pinto, L. and Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE.

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

[Prasad et al., 2017] Prasad, N., Cheng, L.-F., Chivers, C., Draugelis, M., and Engelhardt, B. E. (2017). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint arXiv:1704.06300*.

[Precup, 2000] Precup, D. (2000). Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80.

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

[Qureshi et al., 2017] Qureshi, A. H., Nakamura, Y., Yoshikawa, Y., and Ishiguro, H. (2017). Show, attend and interact: Perceivable human-robot social interaction through neural attention q-network. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1639–1645. IEEE.

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

- [Raghu et al., 2017a] Raghu, A., Komorowski, M., Ahmed, I., Celi, L., Szolovits, P., and Ghassemi, M. (2017a). Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*.
- [Raghu et al., 2017b] Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., and Ghassemi, M. (2017b). Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422*.
- [Rhinehart et al., 2018] Rhinehart, N., McAllister, R., and Levine, S. (2018). Deep imitative models for flexible inference, planning, and control. *arXiv preprint arXiv:1810.06544*.
- [Rich et al., 2010] Rich, C., Ponsler, B., Holroyd, A., and Sidner, C. L. (2010). Recognizing engagement in human-robot interaction. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 375–382. IEEE.
- [Riedmiller, 2005] Riedmiller, M. (2005). Neural fitted q iteration—first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328. Springer.
- [Ritschel et al., 2017] Ritschel, H., Baur, T., and André, E. (2017). Adapting a robot’s linguistic style based on socially-aware reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (ro-man)*, pages 378–384. IEEE.
- [Robinson et al., 2014] Robinson, H., MacDonald, B., and Broadbent, E. (2014). The role of healthcare robots for older people at home: A review. *International Journal of Social Robotics*, 6(4):575–591.
- [Ross et al., 2011] Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Pro-*

ceedings of the fourteenth international conference on artificial intelligence and statistics, pages 627–635.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

[Saerbeck et al., 2010] Saerbeck, M., Schut, T., Bartneck, C., and Janse, M. D. (2010). Expressive robots in education: varying the degree of social supportive behavior of a robotic tutor. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1613–1622. ACM.

[Sallab et al., 2017] Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76.

[Schmitt et al., 2010] Schmitt, A., Polzehl, T., and Minker, W. (2010). Facing reality: Simulating deployment of anger recognition in ivr systems. In *International Workshop on Spoken Dialogue Systems Technology*, pages 122–131. Springer.

[Schroder et al., 2011] Schroder, M., Bevacqua, E., Cowie, R., Eyben, F., Gunes, H., Heylen, D., Ter Maat, M., McKeown, G., Pammi, S., Pantic, M., et al. (2011). Building autonomous sensitive artificial listeners. *IEEE transactions on affective computing*, 3(2):165–183.

[Schulman et al., 2015] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897.

[Shortreed et al., 2011] Shortreed, S. M., Laber, E., Lizotte, D. J., Stroup, T. S., Pineau, J., and Murphy, S. A. (2011). Informing sequential clinical decision-

making through reinforcement learning: an empirical study. *Machine learning*, 84(1-2):109–136.

[Sidner and Dzikovska, 2002] Sidner, C. L. and Dzikovska, M. (2002). Human-robot interaction: engagement between humans and robots for hosting activities. In *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, pages 123–128.

[Sidner et al., 2005] Sidner, C. L., Lee, C., Kidd, C., Lesh, N., and Rich, C. (2005). Explorations in engagement for humans and robots. *arXiv preprint cs/0507056*.

[Silver et al., 2017] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

[Steinfeld et al., 2006] Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., and Goodrich, M. (2006). Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40.

[Sun et al., 2018] Sun, L., Peng, C., Zhan, W., and Tomizuka, M. (2018). A fast integrated planning and control framework for autonomous driving via imitation learning. In *Dynamic Systems and Control Conference*, volume 51913, page V003T37A012. American Society of Mechanical Engineers.

[Sutton, 1991] Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[Tapus et al., 2008] Tapus, A., Țăpuș, C., and Matarić, M. J. (2008). User—robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy. *Intelligent Service Robotics*, 1(2):169.

- [Tesauro, 1994] Tesauro, G. (1994). Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219.
- [Thomas and Brunskill, 2016] Thomas, P. and Brunskill, E. (2016). Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148.
- [Truong et al., 2010] Truong, K. P., Poppe, R., and Heylen, D. (2010). A rule-based backchannel prediction model using pitch and pause information. In *Eleventh Annual Conference of the International Speech Communication Association*.
- [Tseng et al., 2017] Tseng, H.-H., Luo, Y., Cui, S., Chien, J.-T., Ten Haken, R. K., and El Naqa, I. (2017). Deep reinforcement learning for automated radiation adaptation in lung cancer. *Medical physics*, 44(12):6690–6705.
- [van den Berghe et al., 2019] van den Berghe, R., Verhagen, J., Oudgenoeg-Paz, O., Van der Ven, S., and Leseman, P. (2019). Social robots for language learning: A review. *Review of Educational Research*, 89(2):259–295.
- [Voloshin et al., 2019] Voloshin, C., Le, H. M., Jiang, N., and Yue, Y. (2019). Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*.
- [Wang et al., 2018] Wang, L., Zhang, W., He, X., and Zha, H. (2018). Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2447–2456.
- [Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.
- [Weber et al., 2018] Weber, K., Ritschel, H., Aslan, I., Lingensfelder, F., and André, E. (2018). How to shape the humor of a robot-social behavior adaptation based on

- reinforcement learning. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 154–162.
- [Wu et al., 2019] Wu, Y., Tucker, G., and Nachum, O. (2019). Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*.
- [Yngve, 1970] Yngve, V. H. (1970). On getting a word in edgewise. In *Chicago Linguistics Society, 6th Meeting, 1970*, pages 567–578.
- [Yu et al., 2018] Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., and Darrell, T. (2018). Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6.
- [Yurtsever et al., 2020] Yurtsever, E., Lambert, J., Carballo, A., and Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469.
- [Zhang et al., 2019] Zhang, Y., Song, W., Tan, Z., Zhu, H., Wang, Y., Lam, C. M., Weng, Y., Hoi, S. P., Lu, H., Chan, B. S. M., et al. (2019). Could social robots facilitate children with autism spectrum disorders in learning distrust and deception? *Computers in Human Behavior*, 98:140–149.
- [Zhou et al., 2017] Zhou, L., Small, K., Rokhlenko, O., and Elkan, C. (2017). End-to-end offline goal-oriented dialog policy learning via policy gradient. *arXiv preprint arXiv:1712.02838*.