SOCIAL AND AFFECTIVE MACHINE LEARNING

by

Natasha Jaques

M.Sc., University of British Columbia (2014)
B.Sc. Honours, University of Regina (2012)
B.A., University of Regina (2012)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

at the
Massachusetts Institute of Technology
February 2020

©Massachusetts Institute of Technology 2019. All right reserved.

Author . . . .
Program in Media Arts and Sciences
November 18, 2019

Certified by . . . .
Rosalind Picard
Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by . . . .
Tod Machover
Academic Head
Program in Media Arts and Sciences

SOCIAL AND AFFECTIVE MACHINE LEARNING

by

Natasha Jaques

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning on November 18, 2019 in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

Abstract

Social learning is a crucial component of human intelligence, allowing us to rapidly adapt to new scenarios, learn new tasks, and communicate knowledge that can be built on by others. This dissertation argues that the ability of artificial intelligence to learn, adapt, and generalize to new environments can be enhanced by mechanisms that allow for social learning. I propose several novel deep- and reinforcement-learning methods that improve the social and affective capabilities of artificial intelligence (AI), through social learning both from humans and from other AI agents. First, I show how AI agents can learn from the causal influence of their actions on other agents, leading to enhanced coordination and communication in multi-agent reinforcement learning. Second, I investigate learning socially from humans, using non-verbal and implicit affective signals such as facial expressions and sentiment. This ability to optimize for human satisfaction through sensing implicit social cues can enhance human-AI interaction, and guide AI systems to take actions aligned with human preferences. Learning from human interaction with reinforcement learning, however, may require dealing with sparse, off-policy data, without the ability to explore online in the environment – a situation that is inherent to safety-critical, real-world systems that must be tested before being deployed. I present several techniques that enable learning effectively in this challenging setting. Experiments deploying these models to interact with humans reveal that learning from implicit, affective signals is more effective than relying on humans to provide manual labels of their preferences, a task that is cumbersome and time-consuming. However, learning from humans' affective cues requires recognizing them first. In the third part of this thesis, I present several machine learning methods for automatically interpreting human data and recognizing affective and social signals such as stress, happiness, and conversational rapport. I show that personalizing such models using multi-task learning achieves large performance gains in predicting highly individualistic outcomes like human happiness. Together, these techniques create a framework for building socially and emotionally intelligent AI agents that can flexibly learn from each other and from humans.

Thesis Supervisor: Rosalind Picard
Title: *Professor of Media Arts and Sciences*

SOCIAL AND AFFECTIVE MACHINE LEARNING

by

Natasha Jaques

This dissertation/thesis has been approved by the following committee members:

Thesis Reader . . . .

Douglas Eck
Principal Scientist, Research Director
Google Brain

Thesis Reader . . . .

Nando de Freitas
Principal Scientist
DeepMind

Thesis Reader . . . .

Joelle Pineau
Associate Professor of Computer Science
McGill University
Lead Research Scientist
Facebook AI Research

Acknowledgments

It is well known that grad school can be an arduous journey. And of course because I spent the last several years in grad school, I feel compelled to back up that claim with a citation: grad students are more than 3 times as likely to get depressed as the average American (Barreira et al., 2018). While my PhD was definitely trying at times, I was fortunate to have many amazing and caring people to help see me through it.

I am so thankful to my advisor Rosalind Picard for her kindness, wisdom, generosity, insightful feedback, and for genuinely and thoughtfully supporting me in my career. She gave me the freedom to pursue the projects that I was truly passionate about, for which I am especially grateful. I am thankful to Doug Eck for taking a chance and hiring me as a young intern, and continuing to support me ever since – both practically and through many fun, sharp, and engaging conversations. It has been an honour to learn from Nando de Freitas, and I'm grateful to him for asking tough questions that opened up a new research direction for me. I'm thankful to Justine Cassell for her careful attention to detail and astute feedback. Finally, I would like to thank Joelle Pineau for her patience, being generous with her time, and for providing deep technical feedback that was exceptionally valuable to me.

I am incredibly grateful to my parents, Kevin Jaques and Paula Sostorics, for always providing me with love, support, encouragement, and just the right amount of good-humoured teasing. I'm thankful to my brother Sam Jaques for his insight and our math/CS knowledge-exchange sessions, and to my grandmothers Arlene Sostorics and Mavis Jaques for their steady and practical support. I also want to thank my extended family, including Linda Anderson, Bill Dearborn, Colleen and Ken Sostorics, and Tiffany Cassidy, for making me look forward to coming home.

I am so happy to have Max Kleiman-Weiner in my life, not only for his love and support, and the stimulating, scintillating conversations over many delicious meals shared together, but for his invaluable ability to honestly point out the flaws in my reasoning. I am thankful for many years of fascinating conversations with Juliana Cherston, who has been a sounding board for any and every idea, big or small. My PhD would have been significantly more boring if it weren't for Cristian Jara Figueroa; I'm grateful for the years of laughter, jokes, and camaraderie. I am indebted to Andrew Schonhoffer for his many years of love, friendship, support, humour, patience, and for his uncanny ability to find precisely the most comforting thing to say when I am doubting myself.

To my friends in Boston – Juliana Nazare, Bianca Datta, Kelsey Allen, Peter Beshai, Katie Lyle, Eddy Awad, Martin Saveski, Gavin Lund, and David Ramsay – I feel like you have become like an extended family, and I hope we will stay close through the years ahead. To my friends back home, thank you for all the wild adventures, excellent conversation, and fond memories: Mesel Teklemariam, Jordan Rudderham, Jeremy Lane, Chris Paisley, Sam Schonhoffer, Trevor Holloway, Stefani Kary, and Jordan Brass. And to my ML conference crew – Ben Poole, Niru Maheswaranathan, Laurent Dinh, Kyle Kastner, Johanna Hansen, Anna Huang, Tim Cooijimans, Jason Hartford – thank you for making it an absolute pleasure to work in this field.

I have had the privilege of working with many brilliant colleagues, without whom this PhD would not have been possible. Sara Taylor and I spent many nights working late on projects and papers, and it was an honour to learn from her sharp, steady, and curious approach. I am so grateful to have had fun and supportive collaborations with Asma Ghandeharioun, Judy Hanwen Shen, and Caglar Gulcehre, who are all kind, thoughtful, earnest, and wonderful collaborators. I would especially like to thank Matthias Bauer, Fred Bertsch, Vincent Weixuan Chen, Misha Denil, Shane Shixiang Gu, Curtis 'Fjord' Hawthorne, Ed Hughes, Kristy Johnson, Angeliki Lazaridou, Joel Leibo, Dan McDuff, Nadja Rhodes, Adam Roberts, David Rolnick, Akane Sano, Yannick Schroecker, and Eugene Vinitsky for being brilliant and fantastic collaborators and colleagues.

I also want to thank all the friends and colleagues that I have had the pleasure of working with and interacting with during my PhD: Dan Abolafia, Dhaval Adjodah, Letitia Adu-Sarkodie, Guillaume Alain, Jaan Altosaar, Yannis Assael, Aytar Yusuf, Asaf Azaria, Dzmitry Bahdanau, Adrien Baranes, Sam Beller, Ira Blossom, Philemon Brakel, Noam Brown, Serkan Cabi, Andres Campero, Sherol Chen, Shanshan Sunny Chen, Yutian Chen, Brian Cheung, Junyoung Chung, Joseph Cohen, George Dahl, Andrew Dai, Ishita Dasgupta, Artem Dementyev, Sander Dieleman, Vivian Diep, Karthik Dinakar, Mike Dory, Vincent Dumoulin, Jesse Engel, Tom Erez, Niaja Farve, Szymon Fedor, Bjarke Felbo, Craig Ferguson, Chelsea Finn, Carlos Florensa, Jakob Foerster, Lauren Fratamico, Scott Fujimoto, Anna Goldie, Thore Graepel, Scott Greenwald, Klaus Greff, David Ha, Will Hamilton, Jess

Hamrick, Chris Han, Jean Harb, Mary Heanue, Javier Hernandez, Matt Hoffman, Ethan Holly, Keira Horowitz, Alex Irpan, Geoffrey Irving, Caroline Jaffe, Ina Jazic, Noah Jones, Jackie Kay, Yoo Lim Kim, Helen King, Thomas Kipf, Adam Kosiorek, David Krueger, Tejas Kulkarni, Agata Lapedriza, Kai Lash, Tom Le Paine, Katherine Lee, Josh Lovejoy, Ryan Lowe, Nathan Magnus, Mateusz Malinowski, Dandelion Mané, Jennifer McCleary, Kevin McKee, Pedro Mediano, Ehi Nosakhare, Alexander Novikov, Chris Olah, Sageev Oore, Pedro Ortega, Fengjiao Peng, Eric Pennington, Linda Peterson, Tobi Pfaff, Rachel Popa, Alexander Pritzel, Deirdre Quillen, Neil Rabinowitz, Colin Raffel, Maithra Raghu, Scott Reed, Cinjon Resnick, Dan Robinson, Natalia Díaz Rodríguez, Negar Rostamzadeh, Daniel Rozenberg, Ognjen Rudovic, Rebecca Salois, Colin Santangelo, Oliver Saunders-Wilder, Abi See, Bobak Shahriari, Brendan Shillingford, Francis Song, Sam Spaulding, Mark Spear, DJ Strouse, Csaba Szepesvari, Andrea Tacchetti, John Taggart, Yuval Tassa, Cameron Taylor, James Tolentino, Tiffany Turner, Jane Wang, Ziyu Wang, Steven Wheelwright, Victoria Xia, Mimee Xu, Friedemann Zenke, and Sebastian Zepf.

Finally, I am grateful to the funding agencies that have supported me throughout my PhD, including the MIT Media Lab Consortium, Google, DeepMind, OpenAI, and especially the Government of Canada for providing me with NSERC funding – thanks Justin <3.

1 Introduction 31
1.1 Part I - Social learning from other AI agents 35
1.2 Part II - Social and affective learning from humans 36
1.3 Part III - Detecting social and affective states 38
1.4 Related work 39
1.4.1 Affective Computing 39
1.4.2 Human Robot Interaction 40
1.4.3 Reinforcement Learning and Deep Learning 41
1.5 Contributions 43

2 Methods 45
2.1 Machine learning 46
2.1.1 Types of ML problems 46
2.1.2 Supervised learning 48
2.1.3 Generalization 50
2.2 Deep learning 53
2.2.1 Gradient-based learning 55
2.2.2 Neural network architectures 57
2.3 Reinforcement learning 58
2.3.1 Q-learning 60
2.3.2 Policy gradients 62
2.3.3 Actor critic 63
2.4 Conclusion 64

3 Multi-agent social learning via causal influence 101
3.1 Introduction 102
3.2 Sequential Social Dilemmas 104
3.3 Multi-agent RL for SSDs 106
3.4 Basic social influence 107
3.4.1 Experiment I: Basic influence 109
3.5 Influential communication 111
3.5.1 Experiment II: Influential communication 112
3.6 Modeling other agents 115
3.6.1 Experiment III: Modeling other agents 117
3.7 Related work 117
3.8 Details on causal inference 120

3.9 Conclusions and future work 121
3.10 Statement of contributions 122
3.11 Appendix 122
3.11.1 Influence as Mutual Information 122
3.11.2 Additional experiment - Box Trapped 124
3.11.3 Implementation details 125
3.11.4 Additional results 128

4 Learning from implicit human preferences in dialog 133
4.1 Interactive human evaluation of dialog systems 134
4.1.1 Background 135
4.1.2 Related work 137
4.1.3 Knowledge distillation for sentiment and semantic regularization 138
4.1.4 Evaluation methodologies 140
4.1.5 Details about implicit metrics 142
4.1.6 Experiments 144
4.1.7 Conclusions 148
4.2 Way off-policy batch reinforcement learning 149
4.2.1 Background 150
4.2.2 Related Work 152
4.2.3 RL for language generation 154
4.2.4 Methods 154
4.2.5 Traditional RL experiments 158
4.2.6 Batch RL for learning dialog from human feedback 160
4.2.7 Dialog results 162
4.2.8 Conclusion 166
4.3 KL-control for improved sequence generation with RL 167
4.3.1 Recurrent G-learning 167
4.3.2 KL-control for sequence generation 168
4.3.3 Experiment I: Melody generation 168
4.3.4 Experiment II: Computational drug discovery 174
4.3.5 Conclusion and future work 175
4.4 Statement of contributions 176
4.5 Appendix 177
4.5.1 Self-play hybrid metric coefficients 177
4.5.2 Reddit casual conversation corpus details 178
4.5.3 Embedding-based metrics 179
4.5.4 RL post-hoc metrics 179
4.5.5 Interactive evaluation details 180

4.5.6 Website server setup and configuration 182
4.5.7 Hyper-parameter tuning details 183
4.5.8 One-turn evaluation experiment details 185
4.5.9 Additional results 186

5 Learning from human facial expressions for generating sketches 191
5.1 Introduction 192
5.2 Related work 194
5.3 Methods 195
5.3.1 Study design 195
5.3.2 Machine learning techniques 196
5.4 Experiments 197
5.5 Results 198
5.5.1 Facial expression analysis 198
5.5.2 Machine learning results 200
5.6 Conclusions and future work 201
5.7 Statement of contributions 202

6 Modeling bonding in human conversations 203
6.1 Introduction 204
6.2 Related work 205
6.3 User study 206
6.4 Methods 208
6.4.1 Facial expression detection 208
6.4.2 Skeletal joint extraction 208
6.4.3 Machine learning classification 209
6.5 Results 210
6.5.1 Reliability of the bonding scale 210
6.5.2 Designing an agent to promote bonding 211
6.5.3 Qualitative analysis of non-verbal cues 212
6.5.4 Quantitative analysis of non-verbal cues 213
6.5.5 Predicting bonding in novel conversations. 217
6.6 Conclusions and future work 218
6.7 Statement of contributions 219

7 Machine learning for working with human data 221
7.1 Multimodal autoencoder 222

7.1.1 Introduction 222
7.1.2 Related work 224
7.1.3 Mood prediction dataset 225
7.1.4 Method 227
7.1.5 MMAE results 232
7.1.6 Discussion and conclusion 236
7.2 Automatic identification of artifacts in EDA data 237
7.2.1 Introduction 237
7.2.2 Related work 238
7.2.3 Methods 240
7.2.4 Results 244
7.2.5 Conclusion 246
7.2.6 EDA Explorer 246
7.3 Statement of contributions 247

8 Predicting wellbeing with personalized models 251
8.1 Introduction 253
8.2 Related work 257
8.2.1 Multitask Learning 258
8.2.2 Factors related to wellbeing 260
8.3 User study and feature design 261
8.3.1 Physiology 262
8.3.2 Smartphone features 263
8.3.3 Behavioral surveys and extrinsic variables 264
8.3.4 Weather 264
8.3.5 Location 264
8.3.6 Pre-study survey data 266
8.4 Initial experiments 267
8.4.1 Machine learning problem and methods 267
8.4.2 Feature evaluation 269
8.4.3 Discussion and limitations 272
8.5 Personalized multi-task learning 273
8.5.1 Neural Networks 273
8.5.2 Multi-Task Multi-Kernel Learning 275
8.5.3 Hierarchical Bayesian Logistic Regression (HBLR) 278
8.5.4 Multi-task learning experiments 281
8.5.5 Multi-task learning results and discussion 285
8.6 Personalized regression 292
8.6.1 Gaussian Processes for personalized domain adaptation 293

8.6.2 Regression experiments 296
8.6.3 Regression results 297
8.6.4 Discussion 300
8.7 Conclusions 301
8.8 Statement of contributions 303

9 Conclusion 305
9.1 Future work 306

List of Figures

2.1 Data from the iris dataset proposed by Fisher (1936). The x and y axes are two of the features. The left figure shows the true labels for each of the data points, where red points belong to the *setosa* class, pink are *virginica*, and green are *versicolor*. The right figure shows an assignment of each point to a cluster, learned via *k*-means clustering. Taken from Sanchez (2018). 47

2.2 Logistic sigmoid function. 49

2.3 A classification model learns a **decision boundary** (red line) that divides the red and blue classes. Dotted lines show the margin of the decision boundary. Reproduced with permission from Mubaris (2017). 50

2.4 A classification task where the model must separate class 1 (green X's) from class 2 (circles). Three decision boundaries are shown in red. **Underfitting** occurs when the model capacity is not sufficient to accurately partition the data (a). Here, a linear decision boundary is too simple for the curved distribution of the class labels. However, if the model capacity is too great, it may **overfit** the data, perfectly capturing the idiosyncracies of the training data at the expense of generalization error (c). Figures from Nautiyal (2019). 51

2.5 A simple neural network with two hidden layers (in green), and a one-dimensional output layer. 53

2.6 Red areas show regions of parameter space where the loss is high, blue areas show regions where the loss is low. Convex loss functions (a) are easily to optimize, while neural network architectures can lead to complicated loss landscapes (b). Reproduced with permission from Li et al. (2018a). 55

2.7 RNNs have recurrent cells which receive their own output as input on the next timestep (a). We can visualize RNN processing by **unrolling** the network across time (b). Here, we see a common use-case for RNNs: modeling sequences of data such as text. 58

2.8 In reinforcement learning, an agent interacts with the environment by taking actions and receiving rewards. Based on a similar figure in Sutton and Barto (1998). 59

3.1 The two SSD environments, *Cleanup* (top) and *Harvest* (bottom). Agents can exploit other agents for immediate payoff, but at the expense of the long-term collective reward of the group. Reproduced with permission from Hughes et al. (2018). 105
3.2 Schelling diagrams for the two social dilemma tasks show that an individual agent is motivated to defect, though everyone benefits when more agents cooperate. Reproduced with permission from Hughes et al. (2018). 106
3.3 Total collective reward obtained in Experiment 1. Agents trained with influence (red) significantly outperform the baseline and ablated agents. In Harvest, the influence reward is essential to achieve any meaningful learning. 109
3.4 Moments of high influence. Left: the purple influencer gains influence by moving, signaling the presence of an apple (green tiles) outside the yellow influencee's field-of-view (yellow outlined box). Right: the purple influencer stays still, signaling to the pink influencee that no apples have yet appeared. 110
3.5 The communication model has two heads, which learn the environment policy, π_e, and a policy for emitting communication symbols, π_m. Other agents' communication messages $\boldsymbol{m}_{t-1}$ are input to the LSTM. 111
3.6 Total collective reward for deep RL agents with communication channels. Once again, the influence reward is essential to improve or achieve any learning. 113
3.7 Metrics describing the quality of learned communication protocols. The models trained with influence reward exhibit more consistent communication and more coordination, especially in moments where influence is high. 114
3.8 The Model of Other Agents (MOA) architecture learns both an RL policy π_e, and a supervised model that predicts the actions of other agents, $\boldsymbol{a}_{t+1}$. The supervised model is used for internally computing the influence reward. 115
3.9 Total collective reward for MOA models. Again, intrinsic influence consistently improves learning, with the powerful A3C agent baselines not being able to learn. 117
3.10 Causal diagrams of agent k's effect on j's action. Shaded nodes are conditioned on, and we intervene on a_t^k (blue node) by replacing it with counterfactuals. Nodes with a green background must be modeled using the MOA module. Note that there is no backdoor path between a_t^k and s_t in the MOA case, since it would require traversing a collider that is not in the conditioning set. 120
3.11 The *Box trapped* environment in which the teal agent is trapped, and the purple agent can release it with a special *open box* action. 124

3.12 Number of times the *open box* action occurs at each trajectory step over 100 trajectories. 125
3.13 Influence reward over a trajectory in *Box trapped*. An agent gets high influence for letting another agent out of the box in which it is trapped. 125
3.14 Total collective reward obtained by agents trained to optimize for the collective reward, for the 5 best hyperparameter settings with 5 random seeds each. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps. 128
3.15 Total collective reward times equality, $R * (1 - G)$, obtained in all experiments. Error bars show a 99.5% confidence interval (CI) over 5 random seeds, computed within a sliding window of 200 agent steps. Once again, the models trained with influence reward (red) significantly outperform the baseline and ablated models. 130
3.16 Total collective reward over the top 5 hyperparameter settings, with 5 random seeds each, for all experiments. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps. The influence models still maintain an advantage over the baselines and ablated models, suggesting the technique is robust to the hyperparameter settings. 131

4.1 Illustration of the EI regularization (blue) applied to VHRED baseline (red) to enforce encoding sentiment and semantics of an utterance in the Context RNN. The EI regularization can be similarly applied to HRED and VHCR. 139
4.2 Screenshots of our Interactive Evaluation Platform (available at `https://neural.chat`): (a) chat window (left) and first part of the evaluation form (right); (b) second part of the evaluation form (to show all evaluation questions asked). 141
4.3 64 most frequent emojis in the Twitter corpus, in order of frequency. DeepMoji (Felbo et al., 2017) predicts the probability of each emoji given an utterance, and this vector is used for calculating emotion embeddings. 142
4.4 Assigned weights used for reducing the 64-dimensional emotion embedding into a *Sentiment* score. 143
4.5 Correlations between five human metrics and automated metrics (-U: Calculated on user response, -B: bot response, -U/B: between user and bot response, -B/B: between consecutive bot utterances). **Hybrid Metric $\mathbf{M_H}$ -B/B**, our novel self-play based metric, has a higher correlation across all human metrics than any other metric proposed to-date. 147

4.6 One hundred highest vs. lowest quality conversation trajectories; lines: mean, shaded area: 90% confidence intervals, x-axis: conversation turns. (a) Timing of upvote/downvote ratings: A bad first impression impedes overall rating. (b) Participants talk longer and use more words in conversations rated higher. (c) High-quality conversations elicit more positive user sentiment. (d) High-quality conversations are more semantically similar as measured by average word coherence between user query and bot responses. Note that users tend to leave the conversation after strong negative sentiment or semantic dissimilarity. 148

4.7 EI vs. baseline conversation trajectories; lines: mean, shaded area: 90% confidence intervals, x-axis: conversation turns. (a) EI elicits longer responses from users, suggesting that they are more engaged compared to baseline. (b) EI evokes more laughter from users compared to baseline. (c) EI has higher semantic coherence as measured by average word coherence. The same pattern applies to greedy and extrema word coherence. 148

4.8 In this example batch RL problem, the robot's goal is to travel the minimum distance around the black walls to get to the red flag. A trained behavior policy generated the batch data; the probability of each of the states appearing in the batch, $p_{\mathcal{B}}(s)$, is in yellow (white locations are not contained in the batch). If the offline RL policy estimates the value of going *up* or *left* from the start position is high, it will have no way to refine this estimate using the batch data, or learn a good policy in this region of state space. The KL-constraint ensures that the RL policy will stay within the support of the batch data. However, the behavior policy is suboptimal, so using behavior cloning to directly imitate the batch data will result in suboptimal return. Instead, the KL-constrained model can learn to find the optimal policy, which is within the support of the batch. 150

4.9 Comparison of batch RL algorithms for different offline learning conditions. WOP consistently exceeds the performance of Batch Q-learning, Behavioral Cloning (BC), DBCQ, and the Behavior policy used to generate the batch data. Error bars show 95% *CI* of the mean over 50 trials. 159

4.10 KL-divergence of the policy from the prior is lower with KL-control throughout training. Bands show standard deviation. 163

4.11 Z-scored reward. Red metrics were used in training rewards, green are post-hoc. Traditional RL methods like Batch Q exploit simple action-based rewards, like asking questions. In contrast, KL-control methods shift their distribution towards polite, supportive, and cheerful conversation, allowing them to elicit higher *human reward* (blue). 164

4.12 Comparison of top 10 conversation trajectories observed across deployed models, showing the 90% confidence interval of the rewards. 166

4.13 Average prior reward $p(a|s)$ (probability of notes under the MLE prior), obtained by sampling 100 melodies every 100,000 training epochs. The three models are compared to a model trained using only the music theory rewards r_T. This *RL-only* model shows evidence of catastrophic forgetting, diverging to a policy that is random with respect to the prior. In contrast, KL-control alleviates catastrophic forgetting. 171
4.14 Average music theory reward obtained by sampling 100 melodies every 100,000 training epochs. The three models are once again compared to a model trained using only the music theory rewards r_T. 172
4.15 The number of times a melody from each model was selected as most musically pleasing. Error bars reflect the std. dev. of a binomial distribution fit to the binary win/loss data from each model. KL-control models G, Q, and Ψ are significantly preferred over the prior. 173
4.16 The learned coefficients (λ_i) that the hybrid metric (M_H) is comprised of. Using a leave-bot-out method, we observe that the λ_is are stable. The error bars show 90% confidence intervals. 178
4.17 Interactive evaluation chat interface 180
4.18 Conversation with RL models on `https://neural.chat`. 181
4.19 One-turn evaluation interface crowdworkers see. 185
4.20 Correlation matrix showing the relationships between different aspects of interactive human evaluation. We observe a strong correlation across these human ratings. 187
4.21 Correlation matrix showing the relationships between different automated metrics on self-play trajectories and interactive human ratings aggregated on the bot-level. **Postfixes:** -I: Interactive human evaluation, -B: Calculated on bot response, -B/B: Metric applied to self-play on two consecutive bot generated utterances when the bot converses with itself. 187
4.22 Normalized reward scores obtained by models trained with respect to different rewards. 190

5.1 The web app was designed to indicate to users how the model was interpreting their facial expressions in order to learn how to sketch better. 195
5.2 Steps involved in training the LC-GAN facial feedback model. 197
5.3 Scatterplots showing that human ratings of quality are highly correlated with their facial expressions. 198
5.4 The user expresses contentment and amusement at a scribbled sketch, showing the challenging level of noise present in the human facial expression feedback. 199
5.5 Sketches from the LC-GAN lead to significantly more amused expressions and significantly less sadness. 200

5.6 Samples drawn randomly from the cat, crab, and rhinoceros sketch classes, produced by (a) the original Sketch RNN, and (b) the LC-GAN trained on a small amount of social feedback. 201

5.7 Sketches from the LC-GAN are consistently preferred over sketches from the original model. Error bars show the variance of the estimated binomial distribution. 201

6.1 Experimental set-up for the user study, in which participants conversed while facing each other in a standing position, and were monitored with Kinects, lapel microphones, and cameras attached to a microphone stand. 207

6.2 Definitions and examples of the facial actions and emotion expressions coded by Affectiva automated software (McDuff et al., 2016). The numbers correspond to the FACS codes. A similar figure can be found in McDuff (2016) 208

6.3 The skeletal joint positions tracked by the Microsoft Kinect. Each joint has associated X, Y, and Z coordinates. 209

6.4 Five minutes of facial expressions from the conversation with the least bonding, in which the participant's partner fails to respond empathetically, instead smiling and smirking in response to her sadness and anger. 213

6.5 There is a strong degree of synchrony in whole-body distance (Spine Mid Z) for the pair of participants with the highest bonding. 213

6.6 The facial expressions with the largest differences between conversations with high bonding (blue) and conversations with low bonding (red). If the Z-score is below zero, it means the behavior was less frequent in this group's conversations relative to the overall average. Top: participant's expression, bottom: partner's expression. 214

6.7 The heatmap shows the difference in correlation coefficient ($r_{diff} = r_h - r_l$) between conversations with high bonding (r_h) and low bonding (r_l). Blue tiles represent a correlation that is more strongly positive in high bonding conversations, while red represents a correlation more prevalent in low bonding. 215

7.1 Image inpainting with an autoencoder, reproduced from Shcherbakov and Batishcheva (2014) 225

7.2 A basic autoencoder model, which learns to reconstruct X' given input X, after compressing X into the embedding representation Z. 228

7.3 Data loss in traditional supervised learning paradigm 229

7.4 The full feature vector containing 11 modalities. MMAE reconstruction (red) and PCA reconstruction (green) are compared to the original data (black). Areas shaded grey have been masked to produce $\widetilde{X}$. 231

7.5 MMAE reconstruction (red), PCA reconstruction (green), original data (black). As in Figure 7.4, masked data has been shaded grey. 232
7.6 Stress prediction accuracy on the held-out test set as a function of the number of missing modalities. Error bars show 95% confidence intervals. Mood and health showed a similar pattern. 235
7.7 Plot of a typical Skin Conductance Response (SCR), including both shape and length. 239
7.8 A portion of the signal containing artifacts. The raw signal is shown on the left; a 1Hz low-pass filter has been applied to the signal on the right. As is evident, the filter cannot remove the artifact. 240
7.9 A subset of a single participant's data which includes true SCRs and artifacts. The red and grey shading shows epochs labeled as artifact and unsure, respectively. We note that both classifiers label true SCRs as clean signal. 244
7.10 An example of a typical artifact, similar to Fig. 7.8, where the participant removed the sensor. Red and grey shading show where the classifiers labeled the SC data as artifact and questionable, respectively. 245

8.1 Percent of participants sleeping, studying, in extra-curricular activities, and exercising throughout the day. 261
8.2 Features extracted for each non-artifact SCR detected in the SC signal. 262
8.3 SMS frequency over four days with varying mood. The pattern of texting is markedly different on a day when the participant felt sad. 263
8.4 GMM fitted to location data from one participant. Black points are locations visited; the contours mark the probability distribution induced by the model, with darker blue representing more frequently visited locations. 266
8.5 A simplified version of the MTL-NN architecture. Clusters of related people receive specialized predictions from a portion of the network trained with only their data. Shared initial layers extract features relevant to all clusters. 274
8.6 Distribution of self-report labels after discarding the middle 20%. Participants are listed on the x-axis, in order of their average self-report value for that label (each participant is one column). Almost all participants have data from both label classes. 282
8.7 Accuracy for each type of model in the STL, MTL-moods, and MTL-people approaches. Note that the accuracy significantly ($* = p < 0.05$) improves when using multi-tasking over people for each label and for each machine learning method tested. 286

8.8 MTMKL kernel modality weights, reflecting which feature type is most important to the classifier for each task. The ν parameter controls how heavily the task weights are regularized to be similar, and was set by the hyperparameter search. People are highly individualistic in how strongly their mood is affected by different modalities like weather. 287
8.9 Resulting soft clustering (Φ) when predicting the different labels (mood, stress, and health). Each row shows one of the 104 participant's degree of membership in each cluster. We note that there were 4,3, and 17 clusters needed in predicting happiness, stress, and health, respectively. 288
8.10 Distribution of HBLR weights on the *total number of screen on events (5pm-midnight)* feature for each cluster when predicting tomorrow's mood 289
8.11 Example of different weight distributions induced by the soft clustering for 3 different participants in the mood prediction. Participant 3 is almost exclusively in cluster 1, participant 5 is has membership in clusters 0, 1, and 2, and participant 31 is almost exclusively in cluster 2. 289
8.12 Mean feature weights for mood clusters in HBLR model. The positive label is "Happy" so features with positive (negative) mean weights contribute to being more happy (sad) tomorrow. 291
8.13 Mean feature weights for clusters in HBLR model. For stress the positive label is "Calm" so features with positive (negative) mean weights contribute to being more calm (stressed) tomorrow. 292
8.14 Distribution of self-report labels in the data. Students frequently report feeling healthy and stressed. 296
8.15 MAE per person for both the personalized and generic models. For 61 out of 69 participants, MAE is lower with the personalized MTL-NN model. For GPs, personalization is better for 40 out of 69 participants. 298
8.16 NN and MTL-NN predictions for each outcome for a randomly selected subject compared to the ground truth mood report data, which has been sorted by intensity and connected with a trend line. 298
8.17 Predicted mean and variance over the held-out test data as learned by the DA-GP. The lines in red and green represent the reported and predicted scores, respectively. As can be seen from depicted certainty levels (in gray), the predicted values are within one standard deviation (uncertainty) intervals estimated by GP. 299

List of Tables

3.1 Optimal hyperparameter settings for the models in the basic influence experiment. 126
3.2 Optimal hyperparameter settings for the models in the communication experiment. 127
3.3 Optimal hyperparameter settings for the models in the model of other agents (MOA) experiment. 127
3.4 Final collective reward over the last 50 agent steps for each of the models considered. Bolded entries represent experiments in which the influence models significantly outperformed the scores reported in previous work on *inequity aversion* (Hughes et al., 2018) 131

4.1 Single-turn evaluation fails to capture a lack of diversity in a dialog model's responses, as well as its inability to track the conversation and respond in emotionally appropriate ways. We argue multi-turn evaluation is needed to evaluate dialog models, and show that our Emotion+Infersent (EI) models trained on a larger and more diverse corpus, produce better interactive dialog. We present strong evidence that our novel dialog self-play framework combined with psychologically motivated novel automated metrics can accurately estimate quality of a model with respect to its ability to carry out multi-turn conversations. 136
4.2 Mean ratings (from humans) for Baseline and EI (Emotion+Infersent) models for HRED, VHRED, and VHCR architectures with 90% confidence intervals. For 3-factor ANOVA results, see Section 4.1.6.2. 145
4.3 Results of automatic traditional metrics for 1-turn responses of models per context of baseline and EI (Emotion + Infersent) models. PPL: perplexity, KL: KL divergence, Avg: Average, Ext: Extrema, Grd: Greedy 146
4.4 Results from human single-turn evaluation for EI (Emotion+Infersent) vs. baseline models as measured by pairwise comparisons of **Quality**. We follow Park et al. (2018) in highlighting the higher value between wins/losses and reporting 90% confidence intervals. 147

4.5 Interactive human evaluation of techniques for off-policy batch RL. KL-control models significantly out-perform other techniques. Ratings are Likert scale, votes and human reward are z-scores. 162
4.6 Purely reward-maximizing methods like *Batch Q* trivially exploit the reward function by asking a question every turn, and using the maximum number of tokens in every sentence. Such models diverge away from realistic language, using implausible phrases. In contrast, KL-control methods output plausible language by staying close to the prior, but shift to using polite, cheerful language to maximize implicit human rewards. 163
4.7 Interactive human evaluation of WOP trained with different reward functions. Sentiment leads to the highest quality. Learning from manual button presses is not as effective as learning from implicit metrics like sentiment. 165
4.8 Statistics of music theory rule adherence based on 100,000 randomly initialized melodies generated by each model. The top half of the table contains metrics that should be minimized, while the bottom half contains metrics that should increase. Bolded entries represent significant improvements over the MLE baseline. 170
4.9 Statistics of molecule validity and quality based on 100,000 randomly initialized samples. Bolded entries represent significant improvements over the MLE baseline. 175
4.10 Summary table of ratings collected per model. 181
4.11 Hyper-parameters used for different models. 184
4.12 Automatic metrics computed on ablations of the EI models, trained with distillation from only the emotion recognition model (EI_{emo}), the infersent model (EI_{inf}), or receiving emotion and infersent only as input, without knowledge distillation (*input-only*). Whether emotion or semantics provides the most benefit depends on the dataset and the model. 186
4.13 Results from human single-turn evaluation for EI vs. Baseline models for HRED, VHRED, and VHCR models across quality, fluency, relatedness and empathy pairwise comparisons with 90% confidence intervals 188
4.14 Count of ambiguous examples in human one-turn evaluation. 189

5.1 Significant correlations between facial expressions and quality ratings. 198

6.1 Pearson's r correlations between B-WAI and conversation quality. Bolded measures are significant after performing a Bonferroni correction. 211
6.2 Linear regression coefficients for each of the factors in the model. 212

6.3 The skeletal joint features with the highest information gain. All features are significantly correlated with bonding after applying a Bonferroni correction. 217

7.1 Missing data in the SNAPSHOT study. More than half the data are lost when we require that all data sources and the classification label are present. 227
7.2 RMSE for each modality. Bolded entries are significant improvements. We see that the MMAE significantly outperforms PCA in reconstructing missing data, reducing the total RMSE by over 75%. 233
7.3 Mood, health, and stress prediction accuracy on the held-out test set obtained with different approaches for filling missing data. 235
7.4 Number of Epochs in each classifier. 242
7.5 Computed features on both the raw and filtered SC signal, as well as wavelet coefficient features. 242
7.6 Classifier settings and accuracy on the held-out test-set. The binary classifier shows very strong performance in detecting artifacts in new data. 244
7.7 Features Selected for Binary Classification 245

8.1 Correlation matrix for self-reported measures of wellbeing. All correlations are significant at the $p = 0.01$ level. 267
8.2 Features with the highest information for predicting happiness gain for each modality 270
8.3 Initial classification results of naively applying generic machine learning models to classify happiness. Test accuracy ranges from 56-68% across modalities. Optimal classifier and parameter settings are shown for each modality. 271
8.4 Confusion matrix for ensemble classifier 272
8.5 Features selected from each modality for the second set of experiments. There are a total of 21. 284
8.6 Prediction performance (Accuracy and AUC) of the STL, MTL-moods, and MTL-user methods. Bolded entries represent significant improvements over the STL model, indicating that multitasking for personalization is by far the most effective approach. 285
8.7 Post-hoc analysis of the clusters learned for the HBLR mood model, using pre-study personality and trait measures. Bolded entries represent significant differences from the sample average. 290
8.8 Computed pre-study measures for the HBLR stress prediction clusters. Bolded entries represent significant differences from the sample average. 292

8.9 MAE in predicting wellbeing on the held-out test set, for both traditional ML models and personalized models. Bolded entries represent significant improvements over the non-personalized version of the model ($p < .05$). 297
8.10 ICC, a measure of model fit, in predicting wellbeing on the held-out test set. Bolded entries indicate an improvement of at least 50% over the non-personalized model. 299

Statement of Contributions

Material from the following papers was used to create the chapters of this thesis. Each chapter also includes a section at the end crediting specific co-authors for their contributions.

Chapter 3:

- Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P. A., Strouse, D. J., Leibo, J.Z. & de Freitas, N., "Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning", *International Conference on Machine Learning (ICML)*, (2019).

Chapter 4:

- Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., & Picard, R., "Way Off-Policy Batch Deep Reinforcement Learning of Implicit Human Preferences in Dialog", *International Conference on Learning Representations (ICLR) (submitted)*, (2020).

- Ghandeharioun, A.*, Shen, J. H.*, Jaques, N.*, Ferguson, C., Jones, N., Lapedriza, A., & Picard, R., "Approximating Interactive Human Evaluation with Self-Play for Open-Domain Dialog Systems", *Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada (2019).

- Jaques, N., Gu, S., Bahdanau, D., Hernández-Lobato, J. M., Turner, R. E. & Eck, D., "Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control", *International Conference on Machine Learning (ICML)*, Sydney, Australia (2017).

Chapter 5:

- Jaques, N., McCleary, J., Engel, J., Ha, D., Bertsch, F., Eck, D. & Picard, R., "Learning via Social Awareness: Improving a Deep Generative Sketching Model with Facial Feedback", *International Conference on Learning Representations (ICLR) workshop*, (2018).

Chapter 6:

- Jaques, N., McDuff, D., Kim, Y. K. & Picard R., "Understanding and Predicting Bonding in Conversations Using Thin Slices of Facial Expressions and Body Language", *Intelligent Virtual Agents*, Los Angeles, USA (2016).

- Jaques, N., Kim, Y. K. & Picard R., “Personality, Attitudes, and Bonding in Conversations", *Intelligent Virtual Agents*, Los Angeles, USA (2016).

Chapter 7:

- Jaques, N., Taylor, S., Sano, A. & Picard, R., “Multimodal Autoencoder: A Deep Learning Approach to Filling in Missing Sensor Data and Enabling Better Mood Prediction", *International Conference on Affective Computing and Intelligent Interaction (ACII)*, San Antonio, USA (2017).
- Taylor, S.*, Jaques, N.*, Chen, W., Fedor, S., Sano, A. & Picard R., “Automatic identification of artifacts in Electrodermal Activity data", *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy (2015).

Chapter 8:

- Jaques, N., Rudovic, O., Taylor, S., Sano, A. & Picard, R., “Predicting Tomorrow's Mood, Health, and Stress Level using Personalized Multitask Learning and Domain Adaptation", *Proceedings of Machine Learning Research*, 48, 17-33 (2017).
- Taylor, S.*, Jaques, N.*, Nosakhare, E., Sano, A. & Picard, R., “Personalized Multitask Learning for Predicting Tomorrow's Mood, Stress, and Health", *IEEE Transactions on Affective Computing*, (2017).
- Jaques, N.*, Taylor S.*, Nosakhare E., Sano A. & Picard R., “Multitask Learning for Predicting Health, Stress, and Happiness", *Neural Information Processing Systems (NIPS) Workshop on Machine Learning for Healthcare*, Barcelona, Spain (2016).
- Jaques, N.*, Taylor S.*, Azaria, A., Ghandeharioun, A., Sano A. & Picard R., “Predicting students' happiness from physiology, phone, mobility, and behavioral data", *International Conference on Affective Computing and Intelligent Interaction (ACII)*, Xi'an, China (2015).

Finally, I was an author on several papers during my PhD that are not included in the thesis body, but which I include below for reference:

- Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., Luccioni, A., Maharaj, T., Sherwin, E. D., Mukkavilli, S. K., Kording, K. P., Gomes, C., Ng, A. Y., Hassabis, D., Platt, J. C., Creutzig, F., Chayes, J., Bengio, Y., "Tackling Climate Change with Machine Learning", *(Arxiv preprint)*, (2019).

- Jones, N., Jaques, N., Pataranutaporn, P., Ghandeharioun, A., & Picard, R., "Automatic Triage and Analysis of Online Suicide Risk with Document Embeddings and Latent Dirichlet Allocation", *Affective Computing and Intelligence Interaction (ACII) Workshop on Machine Learning for Affective Computing*, (2019).

- Jaques, N., Gu, S., Turner, R. E. & Eck, D., "Tuning Recurrent Neural Networks with Reinforcement Learning", *International Conference on Learning Representations (ICLR) - workshop*, Toulon, France (2016).

- Taylor, S., Jaques, N., E. Nosakhare, Sano, A., Klerman, E. B. & Picard, R., "Importance of Sleep Data in Predicting Next-Day Stress, Happiness, and Health in College Students", *Journal of Sleep and Sleep Disorders Research (suppl_1)*, A294-A295 (2017).

- Roberts, A., Engel, J., Hawthorne, C., Simon, I., Waite, E., Oore, S., Jaques, N., Resnick, C. & Eck, D., "Interactive Musical Improvisation with Magenta", *Neural Information Processing Systems (NIPS) Demo*, Barcelona, Spain (2016).

- Jaques, N., Rich, T., Dinakar, K., Farve, N., Chen, W.V., Maes, P. & Picard, R., "BITxBIT: Encouraging Behavior Change with N=2 Experiments", *Proceedings of the CHI Conference Extended Abstracts on Human Factors*, San Jose, USA (2016).

- Taylor, S., Jaques, N., Sano, A., Azaria, A., Ghandeharioun, A. & Picard, R., "Machine Learning of Sleep and Wake Behaviors to Classify Self-Reported Evening Mood", *Sleep*, Denver, USA (2016).

- Jaques, N.*, Taylor S.*, Sano A. & Picard R., "Multi-task Multi-Kernel Learning for Estimating Individual Wellbeing", *Neural Information Processing Systems (NIPS) Workshop on Multimodal Machine Learning*, Montreal, Canada (2015).

- Xia, V., Jaques, N., Taylor S., Fedor, S. & Picard R., “Active learning for Electrodermal Activity classification", *IEEE Conference on Signal Processing in Medicine and Biology (SPMB)*, Philadelphia, USA (2015).

- Chen, W., Jaques, N., Taylor, S., Sano, A., Fedor, S. & Picard R., “Wavelet-based motion artifact removal for Electrodermal Activity", *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy (2015).

- Sano, A., Yu A.Z., McHill, A.W., Phillips, A.J., Taylor, S., Jaques, N., Czeisler, C. A., Klerman, E. B. & Picard R., “Prediction of happy-sad mood from daily behaviors and previous sleep history", *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy (2015).

- Jaques, N. & Farve, N., “Engaging the workplace with challenges", *International Conference on Persuasive Technologies*, Chicago, USA (2015).

- Jaques, N., Chen, W. V. & Picard, R., “SmileTracker: Automatically and Unobtrusively Recording Smiles and their Context.", *Proceedings of the CHI Conference Extended Abstracts*, Seoul, Korea (2015).

- Sano, A., Phillips, A. J., Yu, A. Z., McHill, A. W., Taylor, S., Jaques, N., Czeisler, C. A., Klerman, E. B. & Picard, R., “Recognizing academic performance, sleep quality, stress level, and mental health using personality traits, wearable sensors and mobile phones", *Wearable and Implantable Body Sensor Networks (BSN)*, Cambridge, USA (2015).

1 Introduction

In the last few years, a series of breakthroughs in machine learning—and particularly deep learning—have led to an exciting surge of progress in artificial intelligence (AI) research. Deep learning (DL) has advanced technologies like automatic speech recognition (Graves et al., 2013), machine translation (Bahdanau et al., 2016), computer vision (Krizhevsky et al., 2012; Girshick et al., 2014; Hariharan et al., 2011; Wei et al., 2016), and traditional AI research (Mnih et al., 2015; Silver et al., 2016; Agrawal, 2018; Finn, 2018). The benefits of DL for basic science (Evans et al., 2018), medicine (Ardila et al., 2019; Jaques et al., 2017a), and sustainability (Rolnick et al., 2019) are beginning to be realized. Machine learning (ML) now permeates the products that we use every day; search engines like Google, recommender systems like Netflix, personal assistants like Siri, etc. The preponderance of such systems is only going to increase as technologies like autonomous vehicles (Huval et al., 2015) become more widespread.

Despite this inspiring progress, and despite the fact that many current ML systems are part of interfaces that attempt to meet human needs, modern ML still has a critical shortcoming: a lack of social intelligence. Personal assistants cannot understand the meaning behind a user's tone of voice, autonomous vehicles cannot effectively model the intentions of other cars or pedestrians, and recommender systems cannot improve by adapting to the user's changing mood and wellbeing over time. Beyond simply improving ML as it is used today, social learning could provide a general mechanism for enhancing the capabilities of AI more broadly. Many authors who study human cognition argue that human social intelligence is a key driving factor behind our unique intelligence (Harari, 2014; Laland, 2017; Henrich, 2015; Kleiman-Weiner, 2018). Research suggests that social learning is responsible for our rapid cognitive evolution, and the cultural evolution of our species, allowing us to achieve unprecedented progress and coordination on a massive scale (Herrmann et al., 2007; Harari, 2014; van Schaik and Burkart, 2011). While current ML models struggle to learn from interactions with other agents or generalize to unfamiliar tasks, humans' social intelligence allows us to quickly and flexibly learn from the behavior of others, and use this to adapt to novel situations.

What if we could create AI with similar social learning abilities, such that it could benefit from the knowledge of expert humans, or even other agents? Previous research into learning from experts has often focused on techniques such as Learning from Demonstration (LfD) or imitation learning, in which an expert demonstrates each step of executing a task (Chernova and Thomaz, 2014; Argall et al., 2009; Abbeel and Ng, 2004; Schaal, 1999). Providing these demonstrations requires a significant degree of human effort. Similarly, reinforcement learning (RL) agents that learn from human guidance have required a human to manually label correct and incorrect task performance (Knox and Stone, 2009; Christiano et al., 2017). As I show in this dissertation, this approach does not scale. In fact, the current successes of ML in classification and supervised learning have depended on large-scale, manually curated datasets with thousands of human-generated labels (Russakovsky et al., 2015). Although humans are providing these signals, we do not refer to this as typical *social learning*, since no human child requires a parent to manually label thousands of cats before she can learn the concept.

The fact that current ML systems still require such an intense degree of explicit supervision slows the development of ML in areas for which plentiful, labeled data and known metrics do not exist. Humans are often unwilling or unable to provide detailed supervision, and human effort is a much more scarce and expensive resource than training in simulation (Agrawal, 2018). This may explain why major successes in deep reinforcement learning are often restricted to simulated environments (such as games), which have clearly specified metrics to optimize (such as points). The limited scope of this type of explicit supervision severely limits the applicability of ML in the real world.

Intrinsic motivations for reinforcement learning (RL) suggest a possible solution to this issue. The idea is to develop an incentive that causes an agent to learn across multiple environments in the absence of explicit supervision (Chentanez et al., 2005; Lenat, 1976; Schmidhuber, 1991). Current research into intrinsic motivation has proposed incentives like curiosity (a drive for novelty) (e.g. Pathak et al. (2017); Schmidhuber (1991)), and empowerment (a drive for the ability to manipulate the environment) (e.g. Capdepuy et al. (2007)). However, if we consider how humans learn, it is evident we are not only motivated by drives for novelty and power.

As social animals, we find interacting with other people highly motivating, and learn a great deal from these interactions (Bandura and Walters, 1977). We enjoy developing relationships, and our ability to do this is aided by our highly developed ability to sense one another's emotions (Kujawa et al., 2014). In fact, there is substantial evidence that social and emotional processing is a key component of our overall cognitive development and intelligence (Damasio, 1994), and that emotion is heavily involved in the learning process (e.g. (Kort et al., 2001)). There is a wealth of research demonstrating that social interaction is not only inherently rewarding for humans, but fundamental to our normal functioning. Social relationships have been found to be important to wellbeing (Reis and Gable, 2003), cognitive performance and mental health (Hawkley and Cacioppo, 2010), and even physical health and recovery from illness (Cohen and Herbert, 1996; House et al., 1988). According to Social Learning Theory (Bandura and Walters, 1977), observing the attitudes and behaviors of others is central to how humans learn both intelligent behavior, and how to adapt to new situations.

Given the fact that social interaction is inherently rewarding for most humans and other animals, incorporating intrinsic[1] social motivations into AI agents could have beneficial implications for their ability to learn and adapt. In this dissertation, I develop ML methods for social learning, which lead to enhanced performance on a variety of tasks. In the first part, I show how AI agents can learn socially from other AI agents in simulations, through modeling the behavior of other agents and learning how their actions affect others. This leads to enhanced coordination and communication in complex multi-agent social dilemmas. The second part presents methods for learning from humans. Rather than requiring humans to explicitly, manually label good performance, I focus on learning from implicit social cues, such as people's facial expressions, or the sentiment they express in text.

Any AI system which is intended to interact with humans could benefit from the ability to sense implicit social cues, because it can use these cues to infer human preferences and therefore cater to those preferences. Building systems that are intrinsically motivated to adhere to human needs is likely to lead to more safe and beneficial AI.

[1] Note that *intrinsic* is not a synonym of *internal*; other people can be intrinsically motivating (Stavropoulos and Carver, 2013).

However, this necessitates creating a system that has enough emotional intelligence to both accurately sense social cues, and use them to infer human mental states and preferences. Therefore, the third part of this dissertation discusses how to use sophisticated machine learning techniques to work with data from humans, and detect not only superficial cues such as facial expressions, but also infer deeper affective states such as stress, happiness, and conversational rapport.

This dissertation unites disparate ideas from Affective[2] Computing, Human Robot Interaction, Machine learning and Reinforcement Learning for the first time. I use automatic emotion recognition techniques to allow AI agents to automatically sense implicit social and emotional feedback, and use it as an intrinsically motivating cue to improve. Because human data are scarce and noisy, I present several methods for working with this data, including pre-training on more readily available data and fine-tuning with human feedback. I also propose new types of intrinsic social motivation for deep RL which improve coordination and communication among multiple agents training in simulation. I investigate several research questions related to how to improve the social and emotional intelligence of AI agents, and study whether this can result in faster learning and better performance. For example, can a generative model produce better art if it is aware of the facial expressions made in response to the samples it generates? Can a conversational AI produce better dialog if it is rewarded when human responses to its utterances have more positive sentiment? Can agents in a multi-agent environment learn socially from the behavior of other agents, and will this help them to learn more coordinated behavior?

Incorporating social and affective intelligence into ML is a promising direction for improving the intelligence, robustness, and adaptability of future AI. Social learning is a core component of human intelligence, allowing us to rapidly adapt to new scenarios and transmit knowledge. Pursuing the development of social and emotional AI will provide the immediate benefit of building AI systems that can collaborate more effectively with humans, and the long-term research benefit of developing AI systems that learn more generally and flexibly. Current AI models suffer from an often deleterious lack of ability to generalize to new situations or benefit from the implicit feedback they receive when deployed. I believe that social learning may provide a path to overcoming this problem.

[2] *Affective* means relating to emotions moods, feelings, and attitudes.

1.1 *Part I - Social learning from other AI agents*

The fact that most ML systems still rely heavily on explicit supervision hampers the development of broadly applicable, generalizable models. Reinforcement Learning (RL) provides a paradigm that allows agents to learn from ongoing interaction with the environment, yet successes in RL have largely been restricted to simplified, simulated domains where correct performance is described through a fully specified reward function such as points in a game. For real-world tasks, such a reward function is unlikely to be available, and designing a hand-engineered reward function can be a time-consuming and error-prone process (Agrawal, 2018). Such reward functions may be vulnerable to trivial exploitation, where the learner is able to maximize reward without actually learning useful skills. In subjective domains, it may not even be possible to design the reward function. What computable metric accurately describes beautiful music?

Deriving rewards from social interaction may help to overcome these problems, since social rewards are inherently dynamic and constantly evolving. As one agent learns and adapts to the environment, the other agent must continually adapt as well. This is clearly obvious in competitive games; as the strategies of one player get more sophisticated, the other player must improve in order to compete. The dynamic, non-stationary nature of multi-agent learning may provide an inherent *autocurriculum* (Leibo et al., 2019), which continually pushes agents to improve (as demonstrated in recent work on competitive hide-and-seek (Baker et al., 2019)). Social rewards are not trivially exploitable; if one agent discovers an exploitative strategy, the other can learn to defend against it. Thus, by making it inherently rewarding for an agent to interact with another agent, it may be possible to engender complex behavior for scenarios in which there are multiple agents sharing a common environment. For this reason, researchers have recently begun to explore social intrinsic motivations for multi-agent systems (Hughes et al., 2018; Peysakhovich and Lerer, 2018).

However, existing multi-agent research makes assumptions that are not realistic for real-world social learning. For example, many works (e.g. Foerster et al. (2016, 2017); Lazaridou et al. (2018); Cao et al. (2018); Hughes et al. (2018)) assume that agents are able to share private information such as rewards, observations, or even gradients during training time. These assumptions are not valid for real-world scenarios with competitive agents, agents with internal intrinsic motivations, or agents developed at different institutions, and most importantly, cannot generalize to learning from humans.

In **Chapter 3**, I present a new type of intrinsic social motivation for multi-agent learning, where agents are motivated to have a causal influence over the actions of another agent (Jaques et al., 2019b). This is related to rewarding the mutual information between agents' actions. This novel form of social empowerment drives agents to learn to communicate and coordinate in complex, multi-agent social dilemma tasks. Unlike previous work in multi-agent reinforcement learning, our agents are able to learn socially from each other, while still training independently. This is accomplished by equipping each agent with a predictive model of the other agents. An agent can thus ask itself counterfactual questions of the form, "What would the other agent do if I had taken this action instead?", and use its internal model to simulate the answer. The ability for agents to understand the causal impact of their actions on other agents and their environment is a promising future direction for enhancing social learning in AI.

1.2 Part II - Social and affective learning from humans

Beyond improving the long-term intelligence and generalizability of AI, social learning mechanisms could provide an immediate benefit to the ML systems deployed today. In order to make these systems as satisfying and useful to people as possible, we want to train ML algorithms to directly optimize for human preferences. Past work that trained RL algorithms based on human preferences required humans to manually label good performance with explicit button presses (Christiano et al., 2017; Knox and Stone, 2009; Brochu, 2010). However, as I will show in this dissertation, asking users to manually report their preferences and feedback is cumbersome and does not scale.

Consider an AI that is intrinsically motivated to produce positive social responses in the humans with whom it interacts, perhaps deployed within a personal assistant such as Alexa or Siri. When a user responds with an angry or frustrated tone, this could act as a negative incentive, training the model not to repeat the action that led to the user's frustration. Rather than requiring the user to manually train the device, such passive sensing of the user's emotional state could allow the model to learn quickly and at scale, enabling human-in-the-loop training without extra human effort. Although implicit human feedback is likely to be noisy and subject to individual differences, it could be used in a wide range of contexts where people interact with ML systems. Crucially, the representations learned by a socially motivated agent are more likely to capture dimensions of the task that are relevant to human behavior and satisfaction.

Finally, some machine learning problems are really questions about human judgement and taste. What does it mean for an image to look more like art, a song to be more moving, or a joke to be more funny? When the objective function is aesthetic preference, it cannot be approximated without access to human input. Using social incentives as a reward could enable training a model to perform these tasks, and others that involve complex social interactions, where no other current technique can make progress.

Thus, Part II of this dissertation explores learning from human interaction; in particular, learning from implicit social and affective cues. Learning from humans is challenging; human data are scarce, so it is important that algorithms work efficiently and effectively with limited data. Further, it may be unwise to continually update the model as it is interacting with human users, because it may begin to learn inappropriate or unsafe behavior. In **Chapter 4** I tackle both of these issues in the context of learning from implicit cues in human conversation. To use the human data as efficiently as possible, I first pre-train a reasonable model using readily available data, and then use transfer learning to fine-tune it with respect to human feedback. To ensure that the model does not learn inappropriate behavior while interacting with humans, I develop novel deep reinforcement learning algorithms for successfully learning from a fixed batch of limited human interaction data, without allowing the model to explore online in the human environment (Jaques et al., 2019a). This ability is needed for any safety-critical real-world RL system that must be tested before being deployed to interact with humans. I show that learning from implicit signals such as sentiment expressed in the text is more effective than relying on explicit, manual feedback. In **Chapter 5**, I use a different transfer learning technique to improve a deep generative sketching model using the facial expressions of humans viewing samples from the model. This was the first work to show that a deep generative model could be improved with implicit cues such as facial expressions (Jaques et al., 2018).

1.3 *Part III - Detecting social and affective states*

Learning from implicit social cues is important because people convey a large volume of information through nonverbal channels (Mehrabian, 2017). Nonverbal communication—i.e., gestures, body language, tone of voice—not only supplements language, but in some cases can actually replace it (Argyle, 1972). Think of how easily we can detect that our friend or partner is ready to leave a party from across a crowded room. Humans are incredibly effective communicators, readily recognizing emotions in others even when nothing is said explicitly. For AI to work well with humans and in human environments, it should be able to read the rich information sent through nonverbal channels.

However, simply detecting cues such as a smiling facial expression may not be enough to determine a person's underlying mental state. Understanding the broader context in which such a cue occurs can be critical to interpreting it (Kosti et al., 2019). An AI that is motivated to satisfy humans may be more safe, but how can it determine when a human is really satisfied? Such a system requires the emotional intelligence to make inferences about humans' underlying and ongoing mental state. In this dissertation, I present several studies which use a variety of data sources to predict people's underlying affective states. In **Chapter** 6, I focus on predicting whether two people having a conversation are bonding with each other; i.e., experiencing rapport or 'chemistry'. I show that ML models can accurately predict bonding using only short, one-minute snippets of facial expressions and body language (Jaques et al., 2016b,a).

Predicting internal affective states is challenging for a number of reasons, one of which is that collecting and working with human data can be exceptionally difficult. Not only is it expensive to collect, but it is often noisy, sparse, or incomplete. In **Chapter** 7, I show how to use machine learning techniques to overcome these problems. I present a method for learning to reliably predict affective outcomes even when whole data sources go missing; a common problem if a participant forgets to wear a physiological sensor, or does not want to provide a particular data source, such as location data (Jaques et al., 2017b). I also show how to use machine learning to accurately detect artifacts in electrodermal activity (EDA), a useful physiological signal for detecting psychological states like stress (Taylor et al., 2015).

Finally, **Chapter** 8 presents the culmination of several years of work into the problem of predicting a person's wellbeing; specifically, their next day happiness, stress, and health (Jaques et al., 2015a,b, 2016c; Taylor et al., 2017). Data are collected primarily from unobtrusive devices such as smartphones or wrist-worn physiological sensors. We show that generic machine learning methods result in disappointingly low performance on this task, despite sophisticated feature extraction efforts (Jaques et al., 2015a). However, by leveraging multi-task learning techniques to personalize these models and therefore account for individual differences, we are able to achieve a more than 20% improvement in prediction accuracy (Jaques et al., 2016c). Analysis of the learned multi-task models reveals that people react to the same stimuli in dramatically different ways, underscoring the need for custom, personalized models (Taylor et al., 2017). The ability to accurately model long-term wellbeing could help future ML systems optimize for human flourishing, rather than trivially optimizing for metrics such as clicks or watch time with destructive consequences (Roose, 2019).

1.4 *Related work*

The methods proposed here are related to a rich history of research in Affective Computing, Human Robot Interaction, Reinforcement Learning, and Deep Learning. This dissertation brings these diverse directions together for the first time, in order to build state-of-the-art AI agents that are motivated to learn socially, and are able to infer complex information from social and affective cues.

1.4.1 *Affective Computing*

The field of Affective Computing includes using computational techniques to recognize, interpret, and process human emotion and affective states, as well as building emotional mechanisms into machines (Picard, 2000). Early research looked at endowing AI agents with emotional intelligence or emotional learning mechanisms. For example, Hyung-il and Picard (2006) propose a modification to the traditional Q-learning function that incorporates valence and arousal. Gadanho and Hallam (2001) incorporate emotions like fear and anger into a robot navigating a physical environment environment. Emotions arise from basic sensations like hunger, pain, and restlessness, driven by not obtaining enough energy, bumping into walls, and not moving for too long, respectively. For a survey of emotions in RL agents, see Moerland et al. (2018).

A plethora of affective computing studies have focused on detecting human emotion, using data such as facial expressions (e.g. Tian et al. (2001)), text (e.g. Shivhare and Khethawat (2012)), electroencephalogram (EEG) (e.g. Ramirez and Vamvakousis (2012)), electrodermal activity (EDA) (e.g. Zangróniz et al. (2017)), electrocardiogram (ECG) (e.g. Valenza et al. (2014)), voice (e.g. Cho and Kato (2011)), and even eye-tracking data (e.g. Jaques (2014)). Existing research tends to stop here, rather than actually using the detected emotions to train the AI to improve. Very recently, a few authors have been able to use the sentiment in language to guide text generation (Shin et al., 2019), and to provide cues to assess whether instructions for a robot were positive or negative (Krening et al., 2017). However, there is very little work in which detected human emotions serve as an intrinsically motivating reward function.

1.4.2 Human Robot Interaction

Although this dissertation is not concerned with training robots, the field of human robot interaction (HRI) has investigated questions related to social learning from humans. Many studies use Learning from Demonstration (LfD) (Chernova and Thomaz, 2014; Argall et al., 2009) or imitation learning (e.g Duan et al. (2017); Finn et al. (2017b); Abbeel and Ng (2004); Schaal (1999)) to teach a robot how to complete a task by having a human expert demonstrate correct performance. However, such approaches are brittle, because the robot may have little guidance on how to complete the task if it strays from the demonstrated method, and obtaining manual demonstrations requires time consuming human effort and thus may not scale (Agrawal, 2018). This dissertation presents techniques which first pre-train a reasonable policy using readily available data, and then fine-tune the policy using implicit human reactions. This approach is more robust and scaleable, and does not require humans to take special steps to train the model.

Beyond imitation learning, several HRI studies have looked at interactive learning with human feedback. Thomaz and Cakmak (2013) build robots that use active learning to ask questions of human trainers. Some of this work incorporates nonverbal cues, in addition to verbal instructions, but relies on heavily engineered task representations in order to learn (Lockerd and Breazeal, 2004; Thomaz and Breazeal, 2007). Kim and Pineau (2016) use inverse RL to learn how humans typically move in crowded environments, and show that training with this model allows a robotic wheelchair to navigate in a more efficient and natural manner. However, although prominent works in HRI have considered the importance of social cues (Breazeal, 2000), often these works take a human-computer interaction (HCI) approach, focusing on engineering robot social signals and measuring human responses to them, rather than developing new mechanisms for improving AI algorithms. For example, Breazeal et al. (2005) create a robot that responds to human gestures with non-verbal social cues and conduct a human evaluation demonstrating that humans prefer this type of social robot, but do not incorporate machine learning or reinforcement learning into the design.

While the field of HRI has long recognized the importance of robots that can recognize human nonverbal social cues, few works have developed new algorithms for learning from these social signals, or agents that are intrinsically motivated to produce positive reactions in humans. Such ideas could apply much more broadly than in the field of robotics, and improve the usefulness of modern RL and deep learning techniques.

1.4.3 *Reinforcement Learning and Deep Learning*

Intrinsic motivation for RL agents is predicated on the idea that rather than wasting time developing complex reward functions for each domain in which we would like to train an RL agent, it may be more important to invest effort in developing a sophisticated, generalizable motivation system that would be applicable across a number of problems (Chentanez et al., 2005; Lenat, 1976; Schmidhuber, 1991). Once experiments have established the usefulness of a given motivation in several tasks, it could be deployed across a number of systems to improve learning. Originally a concept from the psychology literature, intrinsic motivation refers to the desire to perform an activity for the inherent satisfaction or fun that it provides, rather than because of some expectation that it will eventually lead to a desirable outcome. For example, psychologists theorize that humans may find it inherently enjoyable to rapidly learn to perform a task that they initially could not, regardless of whether performing the task will lead to any personal gain (Oudeyer et al., 2008).

There have been several approaches to designing intrinsic motivations for RL. A common thread has been to reward curiosity or novelty by encouraging the agent to discover new or unpredictable stimuli (e.g. Oudeyer et al. (2008)). This can be implemented by rewarding actions which led to high error in predicting future states (e.g. (Still and Precup, 2012)). To avoid curious agents becoming distracted by random effects in the environment, it is possible to learn a representation which only encodes information relevant to taking actions, and reward prediction error in this this representation space (Pathak et al., 2017).

Other approaches focus on encouraging the agent to find knowledge and competence rewarding. Schmidhuber (2010) proposed that an agent should seek to maximize the first derivative of its learning progress in training an internal model of world dynamics. This also combats the problem of being curious about random effects; since the agent will make no progress in predicting random elements such as white noise, it will not be motivated to focus on learning about those parts of the environment. Such an approach has been referred to as a drive for surprise (Mohamed and Rezende, 2015). A similar intrinsic motivator is known as *empowerment*, which can be defined as maximizing the mutual information between a sequence of K actions beginning in state t and the final state s_{t+K} (Capdepuy et al., 2007; Mohamed and Rezende, 2015). Empowerment can encourage an agent to learn complex behavior because it will seek to move to states in which it can exert more control over future states. By optimizing for empowerment, the agent is optimizing for the ability to influence its environment.

Social intrinsic motivation has also been proposed as a way to drive learning (Breazeal et al., 1998). Recently, motivations such as *inequity aversion* (Hughes et al., 2018) have been used to drive cooperation in multi-agent systems. However, as with much of contemporary multi-agent research, this work depends on agents being able to view other agents' rewards, which is not compatible with other agents having unknown intrinsic motivations, or with learning from humans. It is surprising that more deep RL work on intrinsic social motivation has not incorporated human feedback, but perhaps it can be explained by the fact that some deep learning experts feel that "relying on humans for measuring rewards does not scale and is tedious" (p.12) (Agrawal, 2018).

Existing RL work that does incorporate human feedback has largely been restricted to learning from explicit, manually labeled rewards.

For example, Knox and Stone (2009) had humans observe an RL agent playing simple games like Mountain Car, and press a button when they observed the agent taking an action that they would like to reward. Christiano et al. (2017) demonstrated that training a deep RL model with human feedback can lead to better performance in some Atari games. Here, rather than asking humans to reward individual actions, humans were asked to choose which of two videos of action trajectories they preferred. While the ideas presented in these papers serve as useful examples of how to learn from human preferences, both models were trained manually by requiring humans to explicitly indicate feedback through button presses. In contrast, automatically inferring preference through passively observing a user's emotional response requires no additional effort on the part of the user, and could be used in a wide range of contexts where people interact with ML systems.

1.5 Contributions

The goal is to establish that endowing AI agents with social and affective methods for learning can result in better performance across a variety of tasks and environments, and lead to the development of models that are better able to meet human preferences and adapt to new circumstances. In service of this goal, this dissertation makes the following contributions:

- Novel algorithms for multi-agent social learning, leading to enhanced communication and coordination among agents.

- Novel techniques for learning effectively from limited human data with reinforcement learning, even when it is not possible to explore online in the environment.

- Deep learning models that improve through sensitivity to social and affective feedback such as facial expressions.

- Machine learning techniques for effectively dealing with noisy, limited, missing, and sparse human data

- Machine learning models that can detect whether two humans are bonding with each other, based on their implicit social cues.

- Personalized multi-task learning models that can account for individual differences when predicting affective outcomes like stress and happiness, while still learning from data from across the population.

In addition to these conceptual contributions, the above research has also led to the production of several research artifacts, including fourteen peer-reviewed publications (Jaques et al., 2019b,a, 2017a, 2018, 2017b, 2016b,a,c, 2015a, 2017c, 2015b; Ghandeharioun et al., 2019; Taylor et al., 2015, 2017), open-source software to support these projects (see `https://github.com/natashamjaques` and `https://github.com/mitmedialab/`), and an online tool which deploys some of the trained machine learning models to help other researchers (`https://eda-explorer.media.mit.edu/`).

Further progress towards social and affective AI will require integrating in-depth knowledge of affective computing, psychology, machine learning, deep learning, and reinforcement learning, in order to develop models that can effectively infer others' intentions and preferences, learn from their expertise, and coordinate with them in novel tasks and environments. If these efforts are successful, it could potentially lead to the development of a smarter, more functional AI, that is more aligned with our human behavior, goals, and preferences.

2 *Methods*

2.1 Machine learning 46

2.1.1 Types of ML problems 46

2.1.2 Supervised learning 48

2.1.3 Generalization 50

2.2 Deep learning 53

2.2.1 Gradient-based learning 55

2.2.2 Neural network architectures 57

2.3 Reinforcement learning 58

2.3.1 Q-learning 60

2.3.2 Policy gradients 62

2.3.3 Actor critic 63

2.4 Conclusion 64

Machine learning (ML) is an area of artificial intelligence (AI) concerned with systems that can learn from data and improve without being explicitly programmed. Mitchell (1997) formalizes the concept of *learning* as being able to improve performance on a task given new experience with the task. Thus, we can define machine learning algorithms as those that can improve their performance in response to more data or experience. In practice, modern ML often relates to finding patterns in large-scale data. This ability enables us to develop solutions to tasks that are too difficult to be solved with a set of fixed, manually curated instructions written by humans (Goodfellow et al., 2016). Dramatic progress in ML has occurred over the past decade, thanks in large part to deep learning (DL) (Girshick et al., 2014; Girshick, 2015; Lin et al., 2017; Hariharan et al., 2011; He et al., 2017; Cao et al., 2017; Carreira et al., 2016; Wei et al., 2016; Russakovsky et al., 2015; Agrawal, 2018).

Because this thesis will make use of methods from traditional ML, DL, and reinforcement learning (RL), this chapter provides an introduction to these topics. Section 2.1 presents a broad overview of ML and introduces some general, basic models used in later chapters. DL is introduced in Section 2.2, which explains both general concepts behind all deep learning algorithms, as well as basic types of deep learning models. Finally, Section 2.3 introduces RL. The goal of this chapter is not a comprehensive review of these topics, but rather to provide enough background to understand the rest of the thesis. For more in-depth reviews of ML please see Bishop (2006); Murphy (2012), for DL see Goodfellow et al. (2016), and for RL see Sutton and Barto (1998).

2.1 *Machine learning*

Machine learning typically involves learning from a matrix of data, $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, where n is the number of data points, and m is the number of features. A **feature** is a measurable property of the data point. For example, in the classic data set containing examples of different types of iris flowers (Fisher, 1936), one of the features is the *petal length* in cm. Each row of the data matrix $\boldsymbol{X}$ is an example data point, $\boldsymbol{x} \in \mathbb{R}^m$. In the case of the iris dataset, each row would describe the features of a particular flower. We can think of each row as a vector, and each entry of the vector represents the value of the data point for a particular feature. For example, $X_{3,2}$ is the value of feature 2 of example 3.

2.1.1 *Types of ML problems*

One of the most frequent use cases of machine learning is **supervised learning**. In supervised learnig, we are not only given a matrix of data $\boldsymbol{X}$, but each data point x_i also has an associated **label**, y_i. In this case, we would like to learn how to predict the correct label of a new data point the model has never seen before, by learning a mapping $f : X \to Y$, where X is the input space and Y is the output space. When the output or label is a real value, $f : \mathbb{R}^n \to \mathbb{R}$, we call the problem **regression**. An example regression problem is predicting the price of a home based on its features, such as the square footage and the location. When the output is a categorical variable, $f : \mathbb{R}^n \to \{1, ..., k\}$, we call the problem **classification**. An example classification problem would be determining which images contain a picture of a cat, and which contain pictures of dogs. The iris data set is also a classification problem: we seek to predict the type of a particular iris, out of the classes *iris setosa*, *iris virginica*, or *iris versicolor* (as shown in Figure 2.1). In some cases, we may actually wish to learn a probabilistic model which learns the *conditional probability* of the output, $f(x) = p(y|x)$.

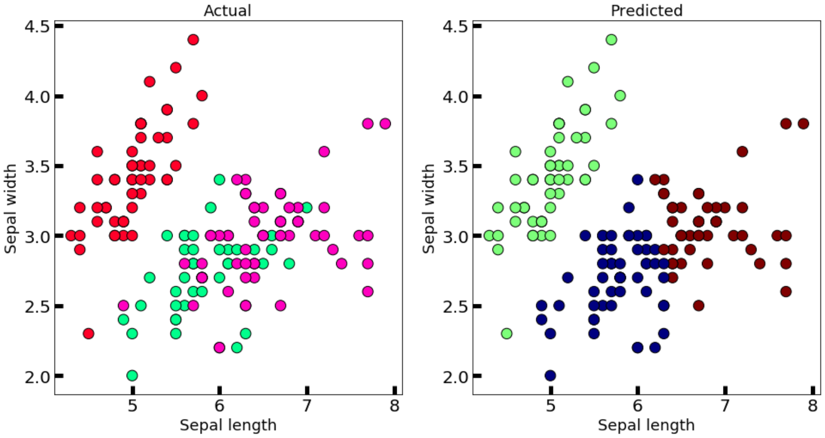


Figure 2.1: Data from the iris dataset proposed by Fisher (1936). The x and y axes are two of the features. The left figure shows the true labels for each of the data points, where red points belong to the *setosa* class, pink are *virginica*, and green are *versicolor*. The right figure shows an assignment of each point to a cluster, learned via k-means clustering. Taken from Sanchez (2018).

In **unsupervised learning**, we are not given any labels Y and must use only the data points of the matrix $\boldsymbol{X}$ to extract patterns from the data. We might seek to estimate the probability distribution that generated the data, or impute missing values (Goodfellow et al., 2016). Often we may wish to group the data into related **clusters**. A well-known clustering algorithm is **k-means clustering** (MacQueen et al., 1967), which learns to assign each data point to one of k clusters such that the distance between all points within the same cluster is minimized. Formally, it partitions the n observations into k sets $\boldsymbol{S} = \{S_1, S_2, ..., S_k\}$ to minimize:

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in S_i} ||\boldsymbol{x} - \boldsymbol{\mu_i}||^2$$

where $\boldsymbol{\mu_i}$ is the mean of the points in cluster S_i. Figure 2.1 (right) shows an example of k-means clustering applied to the iris data. In this example, with only two features and no access to the true labels Y, k-means is not able to learn a partitioning of the data into clusters that matches the true labels.

Not all ML problems fall neatly into the supervised/unsupervised division. Often reinforcement learning (RL) is considered a third major category of ML; we will discuss RL in Section 2.3. Or, a problem might be **semi-supervised**, if some examples have a supervision target y, while others do not. Another category of task is referred to as **structured prediction**. In this case, the output Y is a vector or other data structure with multiple values, including a finite set, a subset of finite sequences, etc. (Goodfellow et al., 2016). Structured prediction problems include translation, scene labeling, and part of sentence tagging. Still other types of ML tasks exist; for a review, see Goodfellow et al. (2016); Murphy (2012).

2.1.2 *Supervised learning*

In the following section, we will dive into supervised learning as a test case that can help explain important ML concepts, such as optimizing a loss function. We will describe several common supervised learning techniques for both regression and classification.

In supervised learning, we are attempting to predict the label y of a data point $\boldsymbol{x}$. Thus, we make a prediction $\hat{y} = f(\boldsymbol{x})$. To learn to make better predictions, we need to optimize some measure of the performance of our model. Typically, this is framed as minimizing a **loss function**, $L(x, y)$.

Consider the case of learning a **linear regression** model:

$$\hat{y} = \boldsymbol{w}^T \boldsymbol{x} \tag{2.1}$$

Here, $\boldsymbol{w}$ is a vector containing the **parameters** of the model, which we will often refer to as **weights** throughout this thesis. Each weight w_i is the coefficient that will be multiplied by feature i. If a feature has a positive weight, it will increase the value of our prediction $\hat{y}$, while features with a negative weight will decrease our predicted $\hat{y}$. Large weights will affect the prediction a lot, while small weights will not.

How can we learn useful values for the weights that will help us make predictions about our data? Consider the following mean squared error loss function:

$$\hat{L}(\boldsymbol{X}, \boldsymbol{y}) = 1/n \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{2.2}$$

This loss sums the difference between our predicted $\hat{y}_i$ and the true y_i over all data points i. Therefore, it computes the total prediction error we make on our dataset, which is a quantity we want to minimize. We can minimize the error by substituting $\boldsymbol{w}^T \boldsymbol{x}$ for $\hat{y}$, and finding the point at which the gradient of the loss with respect to the parameters

$\boldsymbol{w}$ is 0 (i.e. $\nabla_{\boldsymbol{w}}\hat{L} = 0$). Doing this, we can determine that the optimal weight vector is:

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}^T \tag{2.3}$$

For an example of this proof, see Goodfellow et al. (2016).

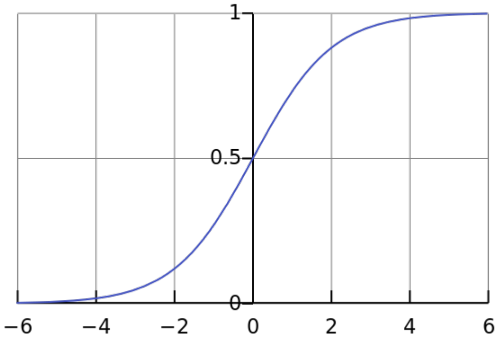


Figure 2.2: Logistic sigmoid function.

Say that we wanted to modify our linear regression model slightly to be able to do binary classification. **Logistic regression** makes use of the **logistic sigmoid** function (plotted in Figure 2.2):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

The logistic function is applied as a non-linear **activation function** to transform the output of $\boldsymbol{w}^T\boldsymbol{x}$. Note that the logistic function in Eq. 2.4 is bounded between 0 and 1. This allows logistic regression to learn the probability that a given example $\boldsymbol{x}$ belongs to a particular class:

$$p(y = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x}) \tag{2.5}$$

To make a decision about which class the example belongs to, we can simply apply a threshold to the learned probability. If $p(y = 1|\boldsymbol{x}) > 0.5$, the example is classified as belonging to class 1. Although logistic regression does not have a simple closed form solution, the loss is convex and easily optimized (Goodfellow et al., 2016).

Classification models learn a **decision boundary**: a plane that separates the input space into regions where the model predicts one class versus the other. Figure 2.3 shows an example decision boundary as a line in a 2D plane. Examples to the right of the line are classified as belonging to the blue class, examples to the left are classified as belonging to the red class. **Support vector machines** (SVMs) are a special type of classification model that seeks to maximize the **margin** between the decision boundary and any of the data points (see

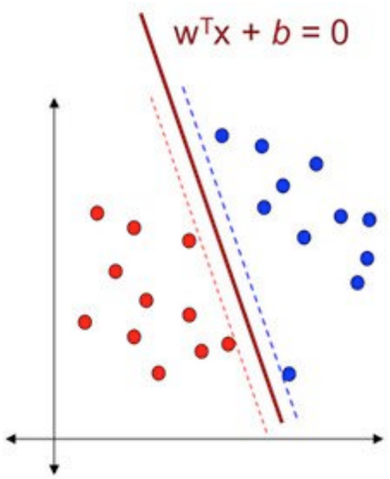


Figure 2.3: A classification model learns a **decision boundary** (red line) that divides the red and blue classes. Dotted lines show the margin of the decision boundary. Reproduced with permission from Mubaris (2017).

Figure 2.3). For a more comprehensive review of SVMs and other common machine learning algorithms, see Bishop (2006); Murphy (2012).

2.1.3 *Generalization*

The central challenge of machine learning is to improve **generalization**, which is the ability to perform well on previously unseen inputs (Goodfellow et al., 2016). If we assume that the true distribution which generates the data (in the world) is p_X^*, then the **generalization error** is:

$$\mathbb{E}_{x \sim p_X^*}[L(f, x)] \tag{2.6}$$

However, there is usually no way to compute this quantity, since we only have access to a finite amount of data D, sampled from p_X^*. Therefore, we can only compute the **empirical loss** (or **empirical risk**) on our n data points:

$$\hat{L}(f, D) = 1/n \sum_{i=1}^{n} L(f, x_i) \tag{2.7}$$

Typically, we divide our data into three subsets: the **training, validation, and test** sets. We attempt to minimize the empirical loss on the training set, and—only once we have finalized our model—measure the empirical loss on the test set as a proxy for the generalization error[1]. If we are able to assume the data are **independent and identically distributed (IID)**—meaning, sampled independently from same

---

[1] It is very important never to **contaminate the test data** by using it to select parameters, features, or model types. Otherwise the test data no longer provides a good proxy for generalization error. We can use the validation set to measure which model settings improve performance.

data-generating distribution p^*_X—we can mathematically study the relationship between training and test error (Dinh, 2018). This is the subject of the field of statistical learning theory.

The difference between the training error and the validation error is an important diagnostic measure that can identify two common phenomena: underfitting and overfitting. **Underfitting** occurs when a model is not able to accurately fit even the training data, and can be diagnosed when the training (or approximation) error is high. The power of the model to accurately fit the data is called the **model capacity**, and if it is too low then underfitting can occur. For example, the capacity of a logistic regression model that is only able to learn a linear classification boundary might be too low to accurately fit a curved function, as shown in Figure 2.4.

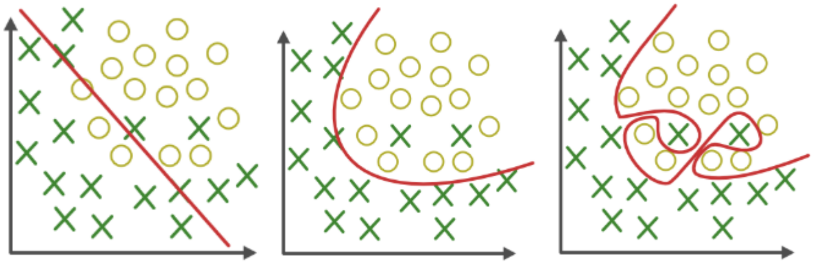

(a) Underfitting (b) Appropriate capacity (c) Overfitting

Figure 2.4: A classification task where the model must separate class 1 (green X's) from class 2 (circles). Three decision boundaries are shown in red. **Underfitting** occurs when the model capacity is not sufficient to accurately partition the data (a). Here, a linear decision boundary is too simple for the curved distribution of the class labels. However, if the model capacity is too great, it may **overfit** the data, perfectly capturing the idiosyncracies of the training data at the expense of generalization error (c). Figures from Nautiyal (2019).

However, if the model capacity is too high, it might begin to **overfit** the data. In this case, it may learn to memorize the idiosyncracies of the training data, rather than learning a generalizable function. This phenomenon is shown in Figure 2.4 (c), where the decision boundary has enough degrees of freedom to include specific training points, but these spurious patterns are unlikely to occur in the test set. We can control the model capacity by altering the **hypothesis space**, which is the set of possible functions that our model is able to learn. For example, we can increase the capacity of a linear regression model to allow it to fit polynomial functions, by changing the model from $\hat{y} = wx + b$ to $\hat{y} = w_2 x^2 + w_1 x + b$ (Goodfellow et al., 2016). Statistical learning theory gives us ways of formally quantifying model capacity. Vapnick-Chervonenkis dimension (**VC dimension**) describes the capacity of a binary classifier as the largest possible training data set size n for which the classifier can label each data point arbitrarily.

The **bias-variance tradeoff** is an important idea that relates to overfitting and underfitting. **Bias error** is a result of incorrect assumptions made by the learning algorithm; for example, assuming the true decision boundary is linear when it is actually curved. High bias can lead to underfitting and therefore high training or approximation error. **Variance** ($\text{Var}(\hat{\boldsymbol{w}})$) describes how much the solution found by the learning algorithm is sensitive to small fluctuations in the training data. If the model has high variance, we expect that the solution may change significantly if we re-sample a new dataset (Goodfellow et al., 2016).

Ideally, we would like to minimize both bias and variance, but often there is a tradeoff: decreasing the bias may lead to higher variance. As we reduce the bias and increase the capacity of the model to fit more complicated data, we tend to increase the variance, and therefore the **generalization gap** (the difference between the generalization error and the training error). We want to find the optimal capacity of the model with the smallest possible generalization gap.

The trainability of a model might limit its capacity as well. If a model is difficult to optimize, it may not be possible to actually find the right function within the family of functions that the model could possibly represent. In this case, the model will have lower **effective capacity**. While we would ideally like a model that is tractable, expressive, and generalizes well (Dinh, 2018), the No Free Lunch Theorem (Wolpert, 1996) states no general purpose algorithm can fulfill all these criteria in every case. Specifically, it states that when averaging over all possible data generating distributions, all algorithms have the same error rate on unseen points (Goodfellow et al., 2016). This means that no one algorithm is universally better, and we must design algorithms with good inductive biases that give good performance in the real world.

Regularization is a way to modify a model in hopes of reducing the generalization error. A common example is **weight decay**, which introduces a penalty on the norm of the parameters of a model, e.g.:

$$J(\boldsymbol{w}) = \hat{L}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{w}) + \lambda \boldsymbol{w}^T \boldsymbol{w} \tag{2.8}$$

Here we see the total loss combines our original loss function (such as mean squared error), with a penalty on the size of the weights. The **hyperparameter** λ controls the strength of the penalty. Most algorithms have hyperparameters that must be tuned by testing the generalization performance using the validation set.

2.2 *Deep learning*

Deep learning refers to a set of techniques for training many different forms of artificial **neural networks** (NNs). As shown in Figure 2.5, a neural network is made up of hidden layers that contain nodes. Each node computes its output based on a set of weights (or parameters) applied to the output of the previous layer. The first layer is applied to the input data $\boldsymbol{x}$. So, node i in the first hidden layer ($\boldsymbol{h}^{(1)}$) computes $\boldsymbol{h}_i^{(1)} = \phi(\boldsymbol{w}_i^{(1)T}\boldsymbol{x})$, where ϕ is an activation function. Note the similarity with the logistic regression model we saw in Eq. 2.5. Essentially, each node is learning its own regression model. Computing the output of all the nodes can be done efficiently using matrix multiplication: $\boldsymbol{h}^{(1)} = \phi(\boldsymbol{W}^{(1)T}\boldsymbol{x})$, where $\boldsymbol{W}^{(1)}$ is the matrix of weights for layer 1.

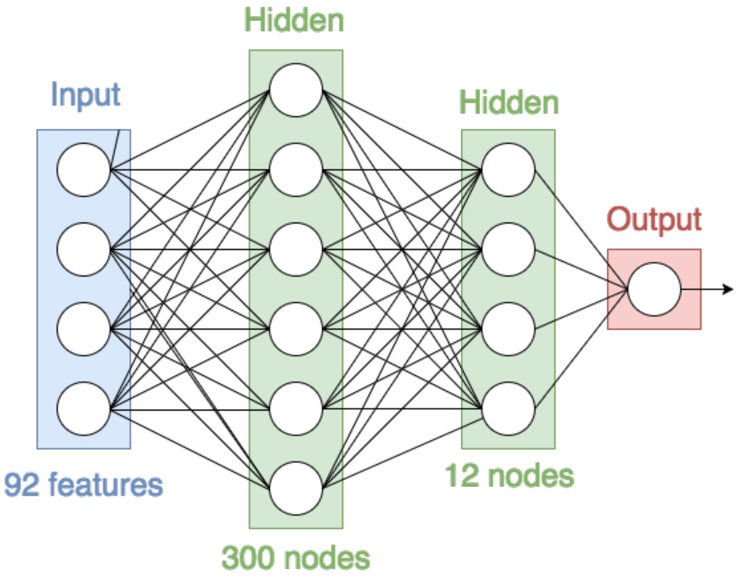


Figure 2.5: A simple neural network with two hidden layers (in green), and a one-dimensional output layer.

Activation functions: The name 'neural' comes from early research which was inspired to (loosely) model the nodes after neurons in the brain. The idea was that each node should aggregate signals from the previous layer, but only 'fire' when these signals became sufficiently strong. This motivates using something like a step-function as an activation function:

$$\phi(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

However, a step-function is difficult to optimize, as we will explain further below. Therefore, activation functions like the logistic sigmoid

(Eq. 2.4) were often used instead. As explained by Goodfellow et al. (2016), modern neural networks more frequently make use of the Rectified Linear Unit (ReLU):

$$\phi_{\text{relu}}(x) = \max(0, x) \tag{2.9}$$

So where does the term deep learning come from? This is due to the ability of neural networks to stack many hidden layers together, with each layer using the output of the previous layer as input, and tranforming it using a non-linear activation function:

$$\boldsymbol{h}^{(m)} = \phi(\boldsymbol{W}^{(m)T}\boldsymbol{h}^{(m-1)}) \tag{2.10}$$

The network learns a series of transformations that can be composed, e.g.: $f^{(3)}(f^{(2)}(f^{(1)}(\boldsymbol{x})))$ (Goodfellow et al., 2016). We describe networks with more layers as having greater **depth**, hence the name.

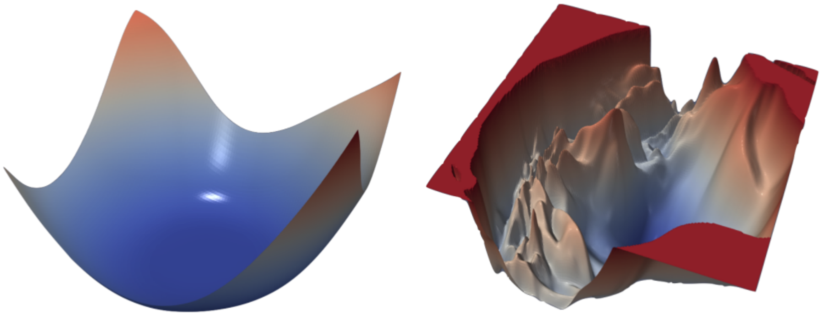
So why go deep? We know from the Universal Approximation Theorem (Cybenko, 1989; Hornik et al., 1990; Barron, 1993) that even a neural network with one hidden layer can approximate any continuous function with arbitrary precision on a bounded data set – as long as that hidden layer is sufficiently large (Goodfellow et al., 2016). Aside from the fact that a single, massive hidden layer is computationally impractical and difficult to optimize, there is a more intuitive reason. Stacking layers allows the network to learn a **hierarchical** set of computations, with later layers building on the features extracted by earlier layers. For example, it has been shown that object detection networks learn to detect simple features such as a line or sphere in early layers, while nodes in the deep layers detect highly abstract entities such as a face, or a cat (Le, 2013). This type of hierarchy allows features in earlier layers to be re-used and composed to create new types of higher level features (i.e both a cat and a face might make use of the sphere feature).

In writing about the nature of intelligence, Hawkins and Blakeslee (2004) argue that hierarchies are extremely important to both the human brain, and to the natural world. Although it is important to emphasize that neural networks are only *loosely* inspired by the human brain, there is nonetheless evidence of similarities between the representations learned by deep neural networks and those of the human brain. For example, networks that process visual input learn a hierarchy of features that is similar to features recognized by the human visual cortex (Kuzovkin et al., 2018).

Deep networks have proven to be extremely effective at compressing information from large-scale data into useful **representations**. The activation of all of the neurons at a particular hidden layer, $h^{(m)}$ can be considered a representation, or an **embedding vector**, or simply an embedding. Given the arguments above, we expect representations learned by the deeper layers of the network to extract useful, abstract information about the data. Ideally, these representations should **disentangle** the factors of variation in the input into informative, reuseable features. Arguably, it is the proficiency of deep learning for **representation learning** that makes it so powerful.

2.2.1 *Gradient-based learning*

As we learned in Section 2.1.2, learning the parameters of a machine learning model involves minimizing a loss function. We can visualize the surface of that function as a **loss landscape**, in which the height of the function indicates the value of the loss function for a particular assignment of parameter values (as shown in Figure 2.6). Traditional ML models often have convex loss functions (e.g. Figure 2.6 (a)) which are easy to optimize. In contrast, due to the number of parameters and non-linearities present in deep neural networks, they have more complicated loss landscapes (e.g. Figure 2.6 (b)). This makes optimization challenging, and precludes the use of traditional methods like linear equation solvers.



(a) Convex loss landscape

(b) Resnet-110 loss landscape

Figure 2.6: Red areas show regions of parameter space where the loss is high, blue areas show regions where the loss is low. Convex loss functions (a) are easily to optimize, while neural network architectures can lead to complicated loss landscapes (b). Reproduced with permission from Li et al. (2018a).

Instead, deep learning relies on iterative, **gradient-based optimization** to find good network parameters by descending the loss landscape in steps, usually starting from a random initialization of the parameters. The simplest example of such an algorithm is **stochastic gradient descent** (SGD). At each step of SGD, we randomly sample a batch of data $\boldsymbol{X_B}$ from the training data, $\boldsymbol{X_B} \sim \boldsymbol{X}_{\text{train}}$. We then compute the gradient of the loss with respect to the parameters given the batch data, $\nabla_{\boldsymbol{w}} L(\hat{\boldsymbol{X_B}}; \boldsymbol{w})$, and move the parameters by a small amount in the direction of the gradient. We can think of this as taking a step down the loss landscape in the direction of steepest descent. A learning rate parameter α controls the size of the step taken. Algorithm 1 gives the procedure for SGD.

Algorithm 1: **Stochastic Gradient Descent (SGD)**

```
Require: $\alpha$ (learning rate)
Require: $\boldsymbol{w_0}$ (initial parameters)
$t \leftarrow 0$ (initialize timestep)
while $\boldsymbol{w_t}$ not converged do
    $\boldsymbol{X_B} \sim \boldsymbol{X}_{\text{train}}$
    $\boldsymbol{g_t} \leftarrow \nabla_{\boldsymbol{w}} L(\hat{\boldsymbol{X}}; \boldsymbol{w})$
    $\boldsymbol{w_{t+1}} \leftarrow \boldsymbol{w_t} - \alpha \boldsymbol{g_t}$
    $t \leftarrow t + 1$
end
```

The gradient with respect to each parameter can be computed efficiently using **backpropagation** (Rumelhart et al., 1985), by recursively applying the chain rule. Frameworks like Tensorflow perform automatic differentiation via backpropagation and computation graphs (Abadi et al., 2015).

The computational cost of SGD is linear in the size of the batch. Using a smaller batch requires less computation and memory, but introduces more noise to the gradient updates (Dinh, 2018). However, the noisy updates of SGD may actually have a regularizing effect (Wilson and Martinez, 2003), which may be important to the empirically demonstrated power of deep neural networks. Modern alternatives to SGD often rely on computing a moving average of the gradient estimates as a form of **momentum** that affects gradient updates (Yu, 2007; Sutskever et al., 2013). Other approaches incorporate an adaptive learning rate. The most frequently used optimization algorithm is Adaptive Moment estimation (AdaM) (Kingma and Ba, 2014).

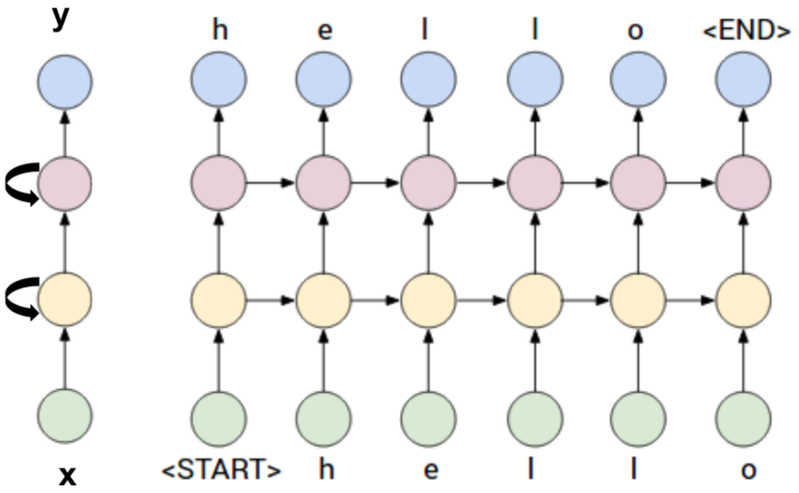
SGD gives an unbiased stochastic approximation of the true gradient, and with mild assumptions will converge to a local minimum even with a batch size of one (Dinh, 2018; Robbins and Monro, 1951; Bottou, 1998). Past working theories of optimization for deep neural networks assumed that local minima presented a signifcant issue for finding good parameter values (e.g. Bengio (2014)). However, recent theoretical research has begun to reveal that under certain assumptions all local minima are actually equivalent to the global minimum (Lu and Kawaguchi, 2017; Kawaguchi, 2016). Similar exciting work has attempted to demonstrate that all optima are actually connected to each other by Bezier curves (Garipov et al., 2018). Further research in optimization has investigated how known techniques such as skip connections can smooth the optimization landscape (Li et al., 2018a; Santurkar et al., 2018), and shown how the common practice of **overparameterizing** neural networks (using more parameters than training data points) can also aid optimization (Arora et al., 2018). Further research into how the techniques engineered by deep learning practioners actually aid optimization is needed.

2.2.2 *Neural network architectures*

The type of network we have been discussing so far is a fully-connected network, where each neuron in a given layer is connected to every neuron in the layer before it. This requires learning a unique set of parameters for every input feature. However, the input may have known regularities that can be better exploited through **parameter sharing**; that is, applying the same weights to multiple parts of the input. This has proven to be an effective strategy, potentially because it improves generalization by limiting the hypothesis space while simultaneously enhancing computational efficiency (Dinh, 2018).

Convolutional neural networks (CNNs) (Fukushima, 1980; LeCun et al., 1998) are one of the most intuitive examples of parameter sharing. These networks contain learned **filters** which are applied across all parts of the input, which is typically an image. In this way, the networks can learn functions which are **translation invariant**. For example, the network can learn a filter to detect a cat, and because it will be applied across many positions in the input image, the network can detect cats in any part of the image.

Rather than sharing parameters across space (as a CNN does for an image), we can also share parameters across time. This is the idea behind **recurrent neural networks** (RNNs), which have neurons (or **cells**) that receive their *own output* from the previous timestep as input (see Figure 2.7). While this can give an RNN a rudimentary form of **memory**, it also exacerbates problems with **vanishing and exploding gradients**. Because computing the gradient depends on multiplying by the same parameter values repeatedly, this can cause the gradients to explode (if the parameter is > 1) or vanish (if the parameter is < 1). Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) help to address this problem by adding an **input, output, and forget gate** to each recurrent cell. These gates allow the network to learn when to update the information in the cell and when to erase it, rather than simply multiplying by the same parameters each time.



(a) RNN w/ 2 recurrent cells

(b) The same RNN unrolled across time. Each column represents the input to the network at a particular timestep.

Figure 2.7: RNNs have recurrent cells which receive their own output as input on the next timestep (a). We can visualize RNN processing by **unrolling** the network across time (b). Here, we see a common use-case for RNNs: modeling sequences of data such as text.

Recently, **transformers** have emerged as an alternative to RNNs (Vaswani et al., 2017). These models make use of an **attention** mechanism (Bahdanau et al., 2016) to summarize inputs of varying lengths based on dynamically changing, learned attention weights. Transformers have been shown to be highly effective at modeling sequences of data, and consequently have led to impressive results in music generation (Huang et al., 2018b) and text generation (Radford et al., 2019).

2.3 *Reinforcement learning*

Reinforcement learning (RL) describes an interactive learning paradigm in which an **agent** interacts with an **environment** by taking **actions** and receiving **rewards** (see Figure 2.8). This interaction takes place across time; at each timestep t, the agent chooses an action

$a_t \in \mathcal{A}$ given the state of the environment $s_t \in \mathcal{S}$, and receives reward $r_t \in \mathbb{R}$. The series of states, actions, and rewards for an episode of interacting with the environment is often termed a **trajectory**, denoted τ. The goal of the agent is to maximize its reward in the long-term (over the trajectory), rather than naïvely taking the action that will lead to the highest reward in the moment. This makes RL a **sequential decision making** problem, distinct from simply predicting future rewards, and other, simpler forms of optimization problems.

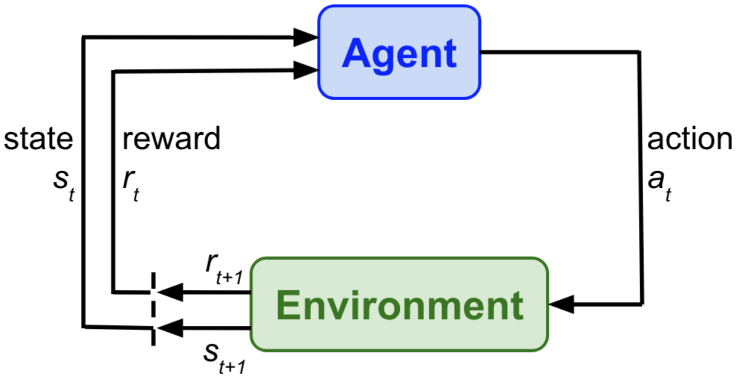


Figure 2.8: In reinforcement learning, an agent interacts with the environment by taking actions and receiving rewards. Based on a similar figure in Sutton and Barto (1998).

In this thesis, we study problems where task performance is specified through a reward function[2]. In other words, there is a function that can assign a high reward value r_t when the agent takes a desirable action a_t in the appropriate state s_t. However, it can be difficult to define a complete reward function for many real-world tasks. This may explain why many prominent successes of RL (e.g. Mnih et al. (2015); Silver et al. (2018)) have been restricted to games, where the reward is specified clearly through the points obtained in the game. It may be possible to overcome the issue of needing to manually design reward functions for each new type of task by instead focusing on **intrinsic motivations**. These are reward functions that may motivate an agent to learn across any number of tasks or environments, such as curiosity. In this thesis, we propose new forms of social intrinsic motivation.

To learn to maximize rewards, the agent learns a policy, π, which maps observations of the state to actions. In deep reinforcement learning, a deep neural network is used to parameterize the policy, such that if the network parameters are $\boldsymbol{\theta}$, the task is to learn $a_t \sim \pi(s_t; \boldsymbol{\theta})$. Often, the agent has limited sensing abilities, or the environment is partially observable, such that the agent sees a restricted view of the state, denoted by o_t or x_t[3]. An agent learns to improve its policy by repeatedly choosing an action and observing the rewards it

[2] Some types of learning which relate to RL do not depend on having a reward function. For example, inverse reinforcement learning and inverse optimal control learn using a sequence of observations, and learning from demonstration (LfD) uses sequences of observation-action pairs.

[3] In this thesis, we may forego using o_t or x_t in favour of just using s_t to represent the agent's view of the state, to simplify notation. In the multi-agent case, we use the notation s_t^j for agent j's partially-observed view of the state.

receives.

However, an agent cannot only seek to maximize rewards, especially not too early in the process of interacting with the environment. How can it know that the current policy it has learned is the best possible policy, without **exploring** more of the environment, and trying actions that it has never tried before? Deciding when to explore by searching for a better policy, and when to **exploit** the current best policy, is a fundamental problem in RL known as the **explore/exploit trade-off**. Frequently, exploration is handled by having the agent act randomly in ϵ proportion of timesteps, and greedily exploiting the best action the rest of the time. This strategy is known as **ϵ-greedy exploration**.

But how can we define the best action to take? We must be able to estimate the expected long-term, future reward; a central problem in RL. Often, the environment is stochastic, meaning that future rewards cannot be predicted exactly. Therefore, we apply a discount factor $\gamma \in [0, 1)$ to rewards received from later timesteps. We can then learn to estimate:

$$V_\pi(s) = \mathbb{E}_{\pi(s_t;\boldsymbol{\theta})}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} \dots | s_t = s] \tag{2.11}$$

$$= \mathbb{E}_{\pi(s_t;\boldsymbol{\theta})}[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s] \tag{2.12}$$

This quantity is called the **value function** or value estimate, and gives the total expected future discounted reward that an agent can expect to obtain when starting in state s_t, and acting according to its policy.

2.3.1 *Q-learning*

Rather than simply learning a value estimate, we often want to learn an **action-value estimate**, $Q_\pi(a, s)$, which gives the expected future reward of starting in state s and taking action a:

$$Q(a, s) = \mathbb{E}_{\pi(s_t;\boldsymbol{\theta})}[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a] \tag{2.13}$$

The action-value function is useful because it informs the agent about which actions are expected to lead to the highest payoff from any given state. Given access to these Q-values, the agent can create a reasonable policy by simply taking the action with the highest $Q(a, s)$[4]:

$$\pi(s) = \max_a Q(a, s) \tag{2.14}$$

[4] Often, we may want to transform the Q-values using a softmax function and then probabilistically sample from them, to allow the agent to continue to explore rather than always greedily selecting the same action. This is known as Boltzmann exploration.

But how can we learn the Q-values? As we improve our policy, we should expect the value estimates to change, since they should describe how much payoff we expect to get following our *updated* policy. This is where the **Bellman equation** comes in handy (Bellman and Kalaba, 1957). The Bellman equation breaks down value estimation using a recursive definition:

$$Q_\pi(a_t, s_t) = r_t + \gamma \mathbb{E}_{p(s_{t+1}|s_t, a_t)}[\max_{a_{t+1}} Q_\pi(a_{t+1}, s_{t+1})] \tag{2.15}$$

The Bellman equation shows that we can describe Q-values in terms of the reward that we actually received at timestep t, r_t, and the future reward we expect to receive starting in the next state that we transitioned to, s_{t+1}. The future reward is estimated using our current best estimates of the Q values for s_{t+1}. Knowing this, we can bootstrap Q-value estimation by repeatedly refining our estimate of $Q(a, s)$ using observed tuples (s_t, a_t, r_t, s_{t+1}) that we get from interacting with the environment. Specifically, we can plug those tuples into the following equation to get the **temporal difference error**:

$$\delta_t = [r_t + \gamma \max_{a_{t+1}} Q_\pi(a_{t+1}, s_{t+1})] - Q_\pi(a_t, s_t) \tag{2.16}$$

The left side of Equation 2.16 is based on plugging our observed tuple into the Bellman equation, while the right term, $Q(a, s)$, is our current best estimate of the Q-value. Thus, TD error gives how far off our current Q estimate is from the reward we just received.

Because the environment is stochastic, it is possible to receive different rewards from taking the same action in the same state, so we do not want to overwrite our previous estimate of $Q(a, s)$ using only r_t and Equation 2.15. Instead, we can create a loss function by squaring the TD error:

$$L(s_t, a_t, r_t, s_{t+1}; \boldsymbol{\theta}) = (r_t + \gamma \max_{a_{t+1}} Q_\pi(a_{t+1}, s_{t+1}) - Q_\pi(a_t, s_t))^2 \tag{2.17}$$

By iteratively optimizing this loss function using the gradient-based learning techniques described in Section 2.2.1, we can refine our Q estimates until they converge to their actual expected value.

An extremely interesting finding that has been replicated repeatedly in the neuroscience literature is that dopamine neurons in the human brain actually encode a reward prediction error signal that is equivalent to the temporal difference error (**?**Hollerman and Schultz, 1998; Schultz, 2007)! These findings suggest that the above RL equations may be a useful and ecologically valid way to learn to estimate future reward based on interactive feedback from the environment.

In **deep Q-learning** (Mnih et al., 2015)—a technique on which several chapters of this thesis are based—we use a deep neural network to estimate $Q(a, s)$ by treating the state s as input, and having an output neuron for each possible action k, which each learn to estimate $Q(a^k, s)$. This is called the **Deep Q-network (DQN)**. However, several modifications are often required to ensure stability in training. The first is to store all observed (s_t, a_t, r_t, s_{t+1}) tuples into an **experience replay buffer**, and randomly sample batches from this buffer to compute the loss in Equation 2.17. The second is to have a **target Q-network** (Van Hasselt et al., 2016) parameterized by $\boldsymbol{\theta_T}$ which is initialized randomly, and only slowly updated so that the parameters move towards the values of the main Q-network. This is because in the early stages of Q-learning, the Q estimates are based on very few samples and can be quite noisy. In fact, the estimates tend to be overly optimistic about future rewards, given the max operator inherent to the Bellman equation. Therefore, the target Q-network is used to provide more conservative estimates of the total expected future reward, such that the loss function becomes:

$$L(s_t, a_t, r_t, s_{t+1}; \boldsymbol{\theta}) = (r_t + \gamma \max_{a_{t+1}} Q_{\pi_{\boldsymbol{\theta_T}}}(a_{t+1}, s_{t+1}) - Q_{\pi_{\boldsymbol{\theta}}}(a_t, s_t))^2 \tag{2.18}$$

2.3.2 *Policy gradients*

While deep Q-learning has led to some impressive results (e.g. Mnih et al. (2015)), some have argued that is more difficult to use and less powerful than alternatives. In fact, a 2016 blog post went so far as to claim that DQNs were "so 2013" (Karpathy, 2016).

A set of alternative approaches are based on the **policy gradients** algorithm. In policy gradients, we have a **policy network** π_θ. As in a DQN, this network inputs the state and has an output for every possible action. However, in this case the output layer is learning a probability distribution over the next action, $\pi_\theta(a_t|s_t) \approx p(a_t|s_t)$[5]. To act, the network simply samples an action: $a_t \sim \pi_\theta(a_t|s_t)$.

Policy gradients is an extremely conceptually simple algorithm. The idea is simply to increase the probability of actions that led to the algorithm receiving a positive reward, and decrease the probability of ones associated with a negative reward. Remember that we always want to maximize the total expected future reward we receive during

[5] To ensure the values of the K output neurons sum to 1.0 and stay within $[0, 1]$—and thus can model a probability distribution—we can apply the **softmax** function:

$$\phi(\boldsymbol{x}_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

a trajectory based on our policy:

$$\mathbb{E}_{\pi_\theta}[r(\tau)] = \mathbb{E}_{\pi_\theta}[\gamma^t r_t] \tag{2.19}$$

The original policy gradients algorithm, REINFORCE (Williams, 1992), simply updates the policy network's parameters based on taking the gradient of this function. Thus, the REINFORCE gradient update is:

$$\nabla_\theta \mathbb{E}_{\pi_\theta}[r(\tau)] = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{T} r_t \nabla_\theta \log \pi_\theta(a_t|s_t)] \tag{2.20}$$

How is the gradient update so simple? Let's go through the steps of the derivation. First, remember that the definition of an expectation is: $\mathbb{E}_x[f(x)] = \sum_x f(x)p(x)$. Substituting $\pi_\theta(a_t|s_t)$ for $p(x)$ and the reward function $r(a_t, s_t) = r_t$ for $f(x)$, we have:

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{\pi_\theta}[r_t] &= \nabla_\theta \sum_{a_t} r_t \pi_\theta(a_t|s_t) && (2.21)\\
&= \sum_{a_t} r_t \nabla_\theta \pi_\theta(a_t|s_t) && (2.22)\\
&= \sum_{a_t} r_t \frac{\pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)} \nabla_\theta \pi_\theta(a_t|s_t) && (2.23)\\
&= \sum_{a_t} r_t \pi_\theta(a_t|s_t) \nabla_\theta \log \pi_\theta(a_t|s_t) && (2.24)\\
&= \mathbb{E}_{\pi_\theta}[r_t \nabla_\theta \log \pi_\theta(a_t|s_t)] && (2.25)
\end{aligned}$$

To get from Eq. 2.23 to Eq. 2.24, we used the definition of the gradient of the log function: $\nabla_\theta \log(x) = \frac{1}{x}\nabla_\theta x$. To get to Eq. 2.25, we simply used the definition of an expectation again.

To use policy gradients in practice, we multiply the gradient $\nabla_\theta \log \pi_\theta(a_t|s_t)$ by both r_t and an indicator function for whether the action a_t was actually taken.

2.3.3 *Actor critic*

In contrast with Q-learning, policy gradients is an on-policy algorithm, meaning that the policy is constantly being updated with the latest experience samples that the agent gets from the environment. Because there tends to be **high variance** in the rewards received, and thus in the gradients, this can lead to instability in the learning process. To solve this problem, a common practice is to subtract a baseline b from the rewards (Williams, 1992):

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[(r_t - b)\nabla_\theta \log \pi_\theta(a_t|s_t)] \tag{2.26}$$

For example, if the baseline is the average reward received, then this function will increase the probability of actions that lead to better than average reward, and decrease the probability of actions that do not.

However, different states are likely to lead to very different rewards, so subtracting the total average might not be very informative. Instead, we could do something more clever using the value estimate of the state, $V(s_t)$. Specifically, we can directly optimize the **advantage function**:

$$\begin{aligned} A(a_t, s_t) &= Q(a_t, s_t) - V(s_t) \qquad (2.27) \\ &= r_t + \gamma V(s_{t+1}) - V(s_t) \qquad (2.28) \end{aligned}$$

Here we obtain Eq. 2.28 because $Q(a_t, s_t) = \mathbb{E}[r_t + \gamma V(s_{t+1})]$.

The advantage function describes how much better the current reward is than what we expect we can get *in this state*. By optimizing it, we increase the probability of actions that lead to better than average reward given the current state. If we substitute the advantage function into policy gradients, we obtain the **advantage actor-critic** (A2C) algorithm (Sutton and Barto, 1998):

$$\begin{aligned} \nabla_\theta J(\theta) &= \mathbb{E}_{\pi_\theta}[(r_t + \gamma V(s_{t+1}) - V(s_t)) \nabla_\theta \log \pi_\theta(a_t|s_t)] \qquad (2.29) \\ &= \mathbb{E}_{\pi_\theta}[A(a_t, s_t) \nabla_\theta \log \pi_\theta(a_t|s_t)] \qquad (2.30) \end{aligned}$$

Optimizing for advantage may seem reasonable, but where did the phrase actor-critic come from? The idea is that the policy is an **actor**, choosing which actions to take. The value estimate serves as a **critic** that can determine whether the action had a good payoff given previous experience. In Chapter 3 of this dissertation, we learn weights for the actor and critic that are both connected to the output of a shared LSTM. We call this having separate **heads** for the actor and critic. We also use modern methods for training A2C asynchronously: asynchronous advantage actor-critic (A3C) (Mnih et al., 2016).

2.4 *Conclusion*

This chapter has briefly reviewed some of the core concepts in Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL). While we have attempted to provide additional background and context for understanding the material in this thesis, there are obviously many more concepts within each of these topics that we were not able to cover here. For much more detailed reviews, consider reading the following textbooks: Bishop (2006); Murphy (2012) for ML, Goodfellow et al. (2016) for DL, and Sutton and Barto (1998) for RL.

Bibliography

Abadi, M. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM.

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*.

Adedokun, O. A. and Burgess, W. D. (2011). Analysis of paired dichotomous data: A gentle introduction to the mcnemar test in spss. *Journal of MultiDisciplinary Evaluation*, 8(17):125–131.

Agarwal, R., Schuurmans, D., and Norouzi, M. (2019). Striving for simplicity in off-policy deep reinforcement learning. *arXiv preprint arXiv:1907.04543*.

Agrawal, P. (2018). *Computational sensorimotor learning*. PhD thesis, University of California, Berkeley.

Aichele, S., Rabbitt, P., and Ghisletta, P. (2016). Think fast, feel fine, live long: A 29-year study of cognition, health, and survival in middle-aged and older adults. *Psychological science*, 27(4):518–529.

Ambady, N. and Rosenthal, R. (1992). Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological bulletin*, 111(2):256.

Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., et al. (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, page 1.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009).

A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

Argyle, M. (1972). Non-verbal communication in human social interaction.

Arora, S., Cohen, N., and Hazan, E. (2018). On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*.

Arroyo, I. et al. (2009). Emotion sensors go to school. In *AIED*, volume 200, pages 17–24.

Astrid, M. et al. (2010). How our personality shapes our interactions with virtual characters-implications for research and development. In *IVA*, pages 208–221.

Avola, D. et al. (2013). Human body language analysis: A preliminary study based on kinect skeleton tracking. In *ICIAP*, pages 465–473.

Azizzadenesheli, K., Brunskill, E., and Anandkumar, A. (2018). Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE.

Bahdanau, D., Brakel, P., Xu, K., Goyal, A., Lowe, R., Pineau, J., Courville, A., and Bengio, Y. (2016). An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. (2019). Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*.

Bandura, A. and Walters, R. H. (1977). Social learning theory.

Barkow, J. H., Akiwowo, A. A., Barua, T. K., Chance, M., Chapple, E. D., Chattopadhyay, G. P., Freedman, D. G., Geddes, W., Goswami, B., Isichei, P., et al. (1975). Prestige and culture: a biosocial interpretation [and comments and replies]. *Current Anthropology*, 16(4):553–572.

Barreira, P., Basilico, M., and Bolotnyy, V. (2018). Graduate student mental health: Lessons from american economics departments. Technical report, Working paper). Retrieved November 29, 2018, from https://scholar. harvard âĂę.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945.

Barton, S. L., Waytowich, N. R., Zaroukian, E., and Asher, D. E.

(2018). Measuring collaborative emergent behavior in multi-agent reinforcement learning. *arXiv preprint arXiv:1807.08663*.

Bauer, G. and Lukowicz, P. (2012). Can smartphones detect stress-related changes in the behaviour of individuals? In *Int. Conf. on Pervasive Comput. and Commun.*, pages 423–426. IEEE.

Baxter, J. (1997). A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1):7–39.

Bellman, R. and Kalaba, R. (1957). Dynamic programming and statistical communication theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43(8):749.

Bengio, Y. (2014). Evolving culture versus local minima. In *Growing Adaptive Machines*, pages 109–138. Springer.

Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Bhatt, A., Argus, M., Amiranashvili, A., and Brox, T. (2019). Cross-norm: Normalization for off-policy td reinforcement learning. *arXiv preprint arXiv:1902.05605*.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. (2012). Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bodie, G. D., St. Cyr, K., Pence, M., Rold, M., and Honeycutt, J. (2012). Listening competence in initial interactions i: Distinguishing between what listening is and what listeners do. *International Journal of Listening*, 26(1):1–28.

Bogin, B., Geva, M., and Berant, J. (2018). Emergence of communication in an interactive world with consistent speakers. *arXiv preprint arXiv:1809.00549*.

Bogomolov, A. et al. (2013). Happiness recognition from mobile phone data. In *Social Computing (SocialCom), 2013 International Conference on*, pages 790–795. IEEE.

Bogomolov, A., Lepri, B., Ferron, M., Pianesi, F., and Pentland, A. (2014). Daily stress recognition from mobile phone data, weather conditions and individual traits. In *ICME*, pages 477–486. ACM.

Bottou, L. (1998). Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142.

Boucsein, W. (2012). *Electrodermal activity*. Springer Science+Business Media, LLC.

Bower, B. et al. (2010). Poor reported sleep quality predicts low positive affect in daily life among healthy and mood-disordered persons. *Journal of sleep research*, 19(2):323–332.

Breazeal, C. et al. (1998). A motivational system for regulating human-robot interaction. In *Aaai/iaai*, pages 54–61.

Breazeal, C., Kidd, C. D., Thomaz, A. L., Hoffman, G., and Berlin, M. (2005). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ international conference on intelligent robots and systems*, pages 708–713. IEEE.

Breazeal, C. L. (2000). *Sociable machines: Expressive social exchange between humans and robots*. PhD thesis, Massachusetts Institute of Technology.

Brebner, J. (1990). Personality factors in stress and anxiety. *Cross-cultural anxiety*, 4:11–19.

Brochu, E. (2010). *Interactive Bayesian optimization: learning user preferences for graphics and animation*. PhD thesis, University of British Columbia.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym.

Butler, E. (2011). Temporal interpersonal emotion systems the âĂIJtiesâĂİ that form relationships. *Pers. and Soc. Psych. Review*, 15(4):367–393.

Caliskan-Islam, A., Bryson, J. J., and Narayanan, A. (2016). Semantics derived automatically from language corpora necessarily contain human biases. *arXiv preprint arXiv:1608.07187*.

Canzian, L. and Musolesi, M. (2015). Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 1293–1304. ACM.

Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., and Clark, S. (2018). Emergent communication through negotiation. *arXiv preprint arXiv:1804.03980*.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings* *of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299.

Capdepuy, P., Polani, D., and Nehaniv, C. L. (2007). Maximization of potential information flow as a universal utility for collective behaviour. In *Artificial Life, 2007. ALIFE'07. IEEE Symposium on*, pages 207–213. Ieee.

Carneiro, D., Novais, P., Augusto, J. C., and Payne, N. (2017). New methods for stress assessment and monitoring at the workplace. *IEEE Transactions on Affective Computing*.

Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. (2016). Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742.

Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

Carver, C. S. and Scheier, M. F. (2011). *Perspectives on personality*. Pearson Higher Ed.

Cheng, H. and Furnham, A. (2003). Personality, self-esteem, and demographic predictions of happiness and depression. *Personality and individual differences*, 34(6):921–942.

Chentanez, N., Barto, A. G., and Singh, S. P. (2005). Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288.

Chernova, S. and Thomaz, A. L. (2014). Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121.

Cho, J. and Kato, S. (2011). Detecting emotion from voice using selective bayesian pairwise classifiers. In *2011 IEEE Symposium on Computers & Informatics*, pages 90–95. IEEE.

Choi, E., Lazaridou, A., and de Freitas, N. (2018). Compositional obverter communication learning from raw visual input. *arXiv preprint arXiv:1804.02341*.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307.

Clark, L., Watson, D., and Mineka, S. (1994). Temperament, personality, and the mood and anxiety disorders. *Journal of abnormal psychology*, 103(1):103.

Cohen, S. et al. (1991). Psychological stress and susceptibility to the common cold. *New England journal of medicine*, 325(9):606–612.

Cohen, S. and Herbert, T. (1996). Health psychology: Psychological factors and physical disease from the perspective of human psychoneuroimmunology. *Annu. rev. of psychology*, 47(1):113–142.

Cohen, S. and Wills, T. (1985). Stress, social support, and the buffering hypothesis. *Psychological bulletin*, 98(2):310.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Crandall, J. W., Oudah, M., Chenlinangjia, T., Ishowo-Oloko, F., Abdallah, S., Bonnefon, J., Cebrián, M., Shariff, A., Goodrich, M. A., and Rahwan, I. (2017). Cooperating with machines. *CoRR*, abs/1703.06207.

Crawford, V. P. and Sobel, J. (1982). Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pages 1431–1451.

Cuperman and Ickes (2009). Big five predictors of behavior and perceptions in initial dyadic interactions. *J. of Pers. and Soc. Psych.*, 97(4):667.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Czeisler, C. et al. (1986). Bright light resets the human circadian pacemaker independent of the timing of the sleep-wake cycle. *Science*, 233(4764):667–671.

Damasio, A. R. (1994). Descartes' error: Emotion, rationality and the human brain.

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics.

Degris, T., White, M., and Sutton, R. S. (2012). Off-policy actor-critic. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 179–186. Omnipress.

Deng, J., Xu, X., Zhang, Z., Frühholz, S., and Schuller, B. (2017).

Universum autoencoder-based domain adaptation for speech emotion recognition. *IEEE Signal Processing Letters*, 24(4):500–504.

Deng, J., Zhang, Z., Marchi, E., and Schuller, B. (2013). Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 511–516. IEEE.

Devlin, S., Yliniemi, L., Kudenko, D., and Tumer, K. (2014). Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 165–172. International Foundation for Autonomous Agents and Multiagent Systems.

Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., et al. (2019). The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Dinh, L. (2018). *Reparameterization in Deep Learning*. PhD thesis, Mila, Université de Montréal.

D'Mello, S. K. and Graesser, A. (2010). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-Adapted Interaction*, 20(2):147–187.

Doberenz, S. et al. (2011). Methodological considerations in ambulatory skin conductance monitoring. *Int. J. of Psychophysiology*, 80(2):87–95.

Dong, W. et al. (2011). Modeling the co-evolution of behaviors and social relationships using mobile phone data. In *Int. Conf. on Mobile and Ubiquitous Multimedia*, pages 134–143. ACM.

Duan, Y., Andrychowicz, M., Stadie, B., Ho, O. J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. (2017). One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098.

Ekman, P. and Friesen, W. (1977). Facial action coding system.

Eleftheriadis, S., Rudovic, O., Deisenroth, M., and Pantic, M. (2016). Gaussian process domain experts for model adaptation in facial behavior analysis. In *CVPR'W*, pages 18–26.

Engel, J., Hoffman, M., and Roberts, A. (2017). Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*.

Ertl, P. and Schuffenhauer, A. (2009). Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8.

Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Zidek, A., Nelson, A., Bridgland, A., Penedones, H., et al. (2018). De novo structure prediction with deeplearning based scoring. *Annu Rev Biochem*, 77:363–382.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. (2018). More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1446–1455.

Fatemi, M., El Asri, L., Schulz, H., He, J., and Suleman, K. (2016). Policy networks with two-stage training for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 101–110.

Fedor, S. and Picard, R. (2014). Ambulatory eda: Comparisons of bilateral forearm and calf locations. 51:S76–S76.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *2017 Conference on Empirical Methods in Natural Language ProcessingConference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ferguson, H. J., Scheepers, C., and Sanford, A. J. (2010). Expectations in counterfactual and theory of mind reasoning. *Language and Cognitive Processes*, 25(3):297–346.

Finn, C. (2018). *Learning to Learn with Gradients*. PhD thesis, UC Berkeley.

Finn, C., Abbeel, P., and Levine, S. (2017a). Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400.

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. (2017b). One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Fleiss, J. L., Cohen, J., and Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5):323.

Foerster, J., Assael, I. A., de Freitas, N., and Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145.

Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. (2018). Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130. International Foundation for Autonomous Agents and Multiagent Systems.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. (2017). Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*.

Forestier, S. and Oudeyer, P.-Y. (2017). A unified model of speech and tool use early development. In *39th Annual Conference of the Cognitive Science Society (CogSci 2017)*.

Fox, R., Pakman, A., and Tishby, N. (2016). Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 202–211. AUAI Press.

Fujimoto, S., Hoof, H., and Meger, D. (2018a). Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1582–1591.

Fujimoto, S., Meger, D., and Precup, D. (2018b). Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900*.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

Gadanho, S. C. and Hallam, J. (2001). Robot learning driven by emotions. *Adaptive Behavior*, 9(1):42–64.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, pages 8789–8798.

Gašić, M., Jurčíček, F., Thomson, B., Yu, K., and Young, S. (2011). On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 312–317. IEEE.

Gauldin (1995). *A practical approach to eighteenth-century counterpoint*. Waveland Pr Inc.

Gelada, C. and Bellemare, M. G. (2019). Off-policy deep reinforcement learning by bootstrapping the covariate shift. *arXiv preprint arXiv:1901.09455*.

Gelman, A. and Hill, J. (2007). Missing-data imputation. *Behavior research methods*, 43(2):310–30.

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Ghandeharioun, A., Shen, J., Jaques, N., Ferguson, C., Jones, N., Lapedriza, A., and Picard, R. (2019). Approximating interactive human evaluation with self-play for open-domain dialog systems. *arXiv preprint arXiv:1906.09308*.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Goldberg, L. R. (1992). The development of markers for the big-five factor structure. *Psychological assessment*, 4(1):26.

Gómez-Bombarelli, R., Duvenaud, D., Hernández-Lobato, J. M., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2016). Automatic chemical design using a data-driven continuous representation of molecules. *arXiv preprint arXiv:1610.02415*.

Gonzales, A. L., Hancock, J. T., and Pennebaker, J. W. (2010). Language style matching as a predictor of social dynamics in small groups. *Communication Research*, 37(1):3–19.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT Press.

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 *AAAI*, pages 3951–3957.

Gratch, J. et al. (2007). Creating rapport with virtual agents. In *Intelligent Virtual Agents*, pages 125–138.

Gratch, J., Okhmatovskaia, A., Lamothe, F., Marsella, S., Morales, M., van der Werf, R. J., and Morency, L.-P. (2006). Virtual rapport. In *Intelligent virtual agents*, pages 14–27. Springer.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Grünerbl, A., Muaremi, A., Osmani, V., Bahle, G., Oehler, S., Tröster, G., Mayora, O., Haring, C., and Lukowicz, P. (2015). Smartphone-based recognition of states and state changes in bipolar disorder patients. *IEEE Journal of Biomedical and Health Informatics*, 19(1):140–148.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. (2016). Continuous Deep Q-Learning with model-based acceleration. In *ICML*.

Guckelsberger, C., Salge, C., and Togelius, J. (2018). New and surprising ways to be mean. adversarial npcs with coupled empowerment minimisation. *arXiv preprint arXiv:1806.01387*.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The J. of Mach. Learning Research*, 3:1157–1182.

Ha, D. and Eck, D. (2017). A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361. JMLR. org.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865.

Hall, M. A. (1998). *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand.

Hancock, B., Bordes, A., Mazare, P.-E., and Weston, J. (2019). Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*.

Harari, Y. N. (2014). *Sapiens: A brief history of humankind*. Random House.

Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., and Malik, J. (2011). Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE.

Hashimoto, C. and Sassano, M. (2018). Detecting absurd conversations from intelligent assistant logs by exploiting user feedback utterances. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 147–156. International World Wide Web Conferences Steering Committee.

Hashimoto, T. B., Zhang, H., and Liang, P. (2019). Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*.

Hawkins, J. and Blakeslee, S. (2004). On intelligence. *New York St. Martins Griffin*, pages 156–8.

Hawkley, L. and Cacioppo, J. (2010). Loneliness matters: a theoretical and empirical review of consequences and mechanisms. *Ann. of Behavioral Medicine*, 40(2):218–227.

Hay, J. (2000). Functions of humor in the conversations of men and women. *Journal of pragmatics*, 32(6):709–742.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Healey, J. and Picard, R. (1998). Digital processing of affective signals. In *Int. Conf. on Acoustics, Speech and Signal Processing*, volume 6, pages 3749–3752. IEEE.

Hedman, E. B. (2010). *In-situ measurement of Electrodermal Activity during Occupational Therapy*. PhD thesis, MIT.

Henrich, J. (2015). *The Secret of Our Success: How culture is driving* *human evolution, domesticating our species, and making us smart*. Princeton University Press, Princeton, NJ.

Hernandez, J., Morris, R. R., and Picard, R. W. (2011). Call center stress recognition with person-specific models. In *ACII*, pages 125–134. Springer.

Herrmann, E., Call, J., Hernàndez-Lloreda, M. V., Hare, B., and Tomasello, M. (2007). Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis. *Science*, 317(5843):1360–1366.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hollerman, J. R. and Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature neuroscience*, 1(4):304.

Hoque, M. and Picard, R. W. (2011). Acted vs. natural frustration and delight: Many people smile in natural frustration. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 354–359. IEEE.

Hornik, K., Stinchcombe, M., and White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560.

Horton, H. (2016). Microsoft deletes 'teen girl' ai after it became a hitler-loving sex robot within 24 hours. In *Telegraph UK*.

Horvath, A. and Greenberg, L. (1989). Development and validation of the working alliance inventory. *Journal of counseling psychology*, 36(2):223.

House, J. et al. (1988). Social relationships and health. *Science*, 241(4865):540–545.

Huang, C., Zaiane, O., Trabelsi, A., and Dziri, N. (2018a). Automatic dialogue generation with expressed emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 49–54.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck,

D. (2018b). Music transformer: Generating music with long-term structure.

Hughes, E., Leibo, J. Z., Phillips, M. G., Tuyls, K., Duéñez-Guzmán, E. A., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K. R., Koster, R., et al. (2018). Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in neural information processing systems (NIPS)*, Montreal, Canada.

Humphrey, N. K. (1976). The social function of intellect. In *Growing points in ethology*, pages 303–317. Cambridge University Press.

Hussain, M. et al. (2011). Affect detection from multichannel physiology during learning sessions with autotutor. In *AIED*, pages 131–138. Springer.

Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., et al. (2015). An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*.

Hyung-il, A. and Picard, R. (2006). Affective-cognitive learning and decision-making: The role of emotions (pdf). In *The 18th European Meeting on Cybernetics and Systems Research (EMCSR 2006)*.

Ireland, M. E., Slatcher, R. B., Eastwick, P. W., Scissors, L. E., Finkel, E. J., and Pennebaker, J. W. (2011). Language style matching predicts relationship initiation and stability. *Psychological science*, 22(1):39–44.

Jaques, N. (2014). *Predicting affect in an Intelligent Tutoring System*. PhD thesis, University of British Columbia.

Jaques, N., Ghandeharioun, A., Shen, J., Ferguson, C., Jones, N., Lapedriza, A., Gu, S., and Picard, R. (2019a). Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

Jaques, N., Gu, S., Bahdanau, D., Hernández-Lobato, J. M., Turner, R. E., and Eck, D. (2017a). Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1645–1654. JMLR. org.

Jaques, N., Kim, Y. L., and Picard, R. (2016a). Personality, attitudes, and bonding in conversations. In *International Conference on Intelligent Virtual Agents*, pages 378–382. Springer.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z., and De Freitas, N. (2019b). Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049.

Jaques, N., McCleary, J., Engel, J., Ha, D., Bertsch, F., Picard, R., and Eck, D. (2018). Learning via social awareness: Improving a deep generative sketching model with facial feedback. *Proceedings of Machine Learning Research*.

Jaques, N., McDuff, D., Kim, Y. L., and Picard, R. (2016b). Understanding and predicting bonding in conversations using thin slices of facial expressions and body language. In *International Conference on Intelligent Virtual Agents*, pages 64–74. Springer.

Jaques, N., Taylor, S., Azaria, A., Ghandeharioun, A., Sano, A., and Picard, R. (2015a). Predicting students' happiness from physiology, phone, mobility, and behavioral data. In *Affective computing and intelligent interaction (ACII), 2015 international conference on*, pages 222–228. IEEE.

Jaques, N., Taylor, S., Nosakhare, E., Sano, A., and Picard, R. (2016c). Multi-task learning for predicting health, stress, and happiness. In *NIPS Workshop on Machine Learning for Healthcare, Barcelona, Spain*.

Jaques, N., Taylor, S., Sano, A., and Picard, R. (2015b). Multi-task, multi-kernel learning for estimating individual wellbeing. In *Proc. NIPS Workshop on Multimodal Machine Learning, Montreal, Quebec*, volume 898.

Jaques, N., Taylor, S., Sano, A., and Picard, R. (2017b). Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *Proc. International Conference on Affective Computing and Intelligent Interaction (ACII), San Antonio, Texas*.

Jaques, N., Taylor, S., Sano, A., Picard, R., et al. (2017c). Predicting tomorrow's mood, health, and stress level using personalized multitask learning and domain adaptation. In *IJCAI 2017 Workshop on Artificial Intelligence in Affective Computing*, pages 17–33.

Jiang, N. and Li, L. (2016). Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661.

Jin, F. and Sun, S. (2008). Neural network multitask learning for traffic flow forecasting. In *IEEE Int'l Joint Conf. on Neural Networks*, pages 1897–1901. IEEE.

John, O. P. and Srivastava, S. (1999). The big five trait taxonomy:

History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research*, 2(1999):102–138.

Kahl, S. and Kopp, S. (2015). Modeling a social brain for interactive agents: integrating mirroring and mentalizing. In *IVA*, pages 77–86.

Kahn, G., Villaflor, A., Pong, V., Abbeel, P., and Levine, S. (2017). Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*.

Kakade, S. M. (2002). A natural policy gradient. In *Advances in neural information processing systems (NIPS)*, volume 14, pages 1531–1538.

Kandemir, M. et al. (2014). Multi-task and multi-view learning of user state. *Neurocomputing*, 139:97–106.

Kapoor, A. and Picard, R. W. (2005). Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 677–682. ACM.

Kappeler-Setz, C. et al. (2013). Towards long term monitoring of electrodermal activity in daily life. *Pers. ubiquit. comput.*, 17(2):261–271.

Kappen, H. J., Gómez, V., and Opper, M. (2012). Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182.

Karpathy, A. (2016). Deep reinforcement learning: Pong from pixels.

Kawaguchi, K. (2016). Deep learning without poor local minima. In *Advances in neural information processing systems*, pages 586–594.

Keller, A. et al. (2012). Does the perception that stress affects health matter? the association with health and mortality. *Health Psychology*, 31(5):677.

Kim, B. and Pineau, J. (2016). Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8(1):51–66.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kisch, J. et al. (2005). Aspects of suicidal behavior, depression, and treatment in college students: results from the spring 2000 national college health assessment survey. *Suicide and Life-Threatening Behavior*, 35(1):3–13.

Kleiman-Weiner, M. (2018). *Computational foundations of human social intelligence*. PhD thesis, Massachusetts Institute of Technology.

Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., and Tenenbaum, J. B. (2016). Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*.

Klerman, G. and Weissman, M. (1989). Increasing rates of depression. *JAMA*, 261(15):2229–2235.

Klimstra, T. et al. (2011). Come rain or come shine: individual differences in how weather affects mood. *Emotion*, 11(6):1495.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: A universal agent-centric measure of control. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 128–135. IEEE.

Knox, W. B. and Stone, P. (2009). Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM.

Kocielnik, R. et al. (2013). Smart technologies for long-term stress monitoring at work. In *Comput.-Based Medical Syst.*, pages 53–58. IEEE.

Koelstra, S. et al. (2012). Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31.

Kohler, C. et al. (2004). Differences in facial expressions of four universal emotions. *Psychiatry research*, 128(3):235–244.

Koldijk, S., Neerincx, M. A., and Kraaij, W. (2017). Detecting work stress in offices by combining unobtrusive sensors. *IEEE Transactions on Affective Computing*, PP(99):1–1.

Kort, B., Reilly, R., and Picard, R. W. (2001). An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Proceedings IEEE International Conference on Advanced Learning Technologies*, pages 43–46. IEEE.

Kosti, R., Alvarez, J., Recasens, A., and Lapedriza, A. (2019). Context based emotion recognition using emotic dataset. *IEEE transactions on pattern analysis and machine intelligence*.

Krening, S., Harrison, B., Feigh, K. M., Isbell, C. L., Riedl, M., and Thomaz, A. (2017). Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kujawa, A., Dougherty, L., Durbin, C. E., Laptook, R., Torpey, D., and Klein, D. N. (2014). Emotion recognition in preschool children: Associations with maternal depression and early parenting. *Development and psychopathology*, 26(1):159–170.

Kumar, A., Fu, J., Tucker, G., and Levine, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*.

Kuzovkin, I., Vicente, R., Petton, M., Lachaux, J.-P., Baciu, M., Kahane, P., Rheims, S., Vidal, J. R., and Aru, J. (2018). Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Communications biology*, 1(1):107.

Laland, K. N. (2017). *Darwin's unfinished symphony : how culture made the human mind / Kevin N. Laland*. Princeton University Press Princeton.

Lane, N. et al. (2010). A survey of mobile phone sensing. *Commun. Magazine, IEEE*, 48(9):140–150.

Lazaridou, A., Hermann, K. M., Tuyls, K., and Clark, S. (2018). Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Leibo, J. Z., Hughes, E., Lanctot, M., and Graepel, T. (2019). Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742*.

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. (2017). Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents* *and MultiAgent Systems*, pages 464–473. International Foundation for Autonomous Agents and Multiagent Systems.

Lenat, D. B. (1976). Am: An artificial intelligence approach to discovery in mathematics as heuristic search. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018a). Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., and Dolan, B. (2016a). A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 994–1003.

Li, J. and Jurafsky, D. (2017). Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209.

Li, J., Miller, A. H., Chopra, S., Ranzato, M., and Weston, J. (2016b). Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016c). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Li, J., Wang, X., and Hovy, E. (2014). What a nasty day: Exploring mood-weather relationship from twitter. In *Int'l Conf. on Info. and Knowledge Management*, pages 1309–1318. ACM.

Li, Z., Kiseleva, J., and de Rijke, M. (2018b). Dialogue generation: From imitation learning to inverse reinforcement learning. *arXiv preprint arXiv:1812.03509*.

LiKamWa, R. et al. (2013). Moodscope: building a mood sensor from smartphone usage patterns. In *Int. Conf. on Mobile systems, applications, and services*, pages 389–402. ACM.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Liu, B. and Lane, I. (2017). Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.

Liu, B., Singh, S., Lewis, R. L., and Qin, S. (2014). Optimal rewards for cooperative agents. *IEEE Transactions on Autonomous Mental Development*, 6(4):286–297.

Liu, B., Tür, G., Hakkani-Tür, D., Shah, P., and Heck, L. (2018). Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069.

Liu, B. and Vasconcelos, N. (2015). Bayesian model adaptation for crowd counts. In *ICCV*, pages 4175–4183.

Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. (2019). Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*.

Lizier, J. T. and Prokopenko, M. (2010). Differentiating information transfer and causal effect. *The European Physical Journal B*, 73(4):605–615.

LLC, T. D. S. C. (2016). Dark sky forecast api.

Lockerd, A. and Breazeal, C. (2004). Tutelage and socially guided robot learning. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 4, pages 3475–3480. IEEE.

Lowe, R., Noseworthy, M., Serban, I. V., Angelard-Gontier, N., Bengio, Y., and Pineau, J. (2017a). Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1116–1126.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. (2017b). Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390.

Lu, H. and Kawaguchi, K. (2017). Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*.

MacKay, D. J. (1992). The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Mazare, P.-E., Humeau, S., Raison, M., and Bordes, A. (2018). Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779.

McDuff, D. (2016). Discovering facial expressions for states of amused, persuaded, informed, sentimental and inspired. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 71–75. ACM.

McDuff, D. et al. (2016). Affdex sdk: A cross-platform real-time multi-face expression recognition toolkit. In *CHI*, pages 3723–3726. ACM.

Meeren, H., van Heijnsbergen, C., and de Gelder, B. (2005). Rapid perceptual integration of facial expression and emotional body language. *Proc. of the National Academy of Sci. of the USA*, 102(45):16518–16523.

Mehrabian, A. (2017). *Nonverbal communication*. Routledge.

Melis, A. P. and Semmann, D. (2010). How is human cooperation different? *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 365(1553):2663–2674.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miotto, R., Li, L., Kidd, B. A., and Dudley, J. T. (2016). Deep patient:

An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, pages 236–244.

Mitchell, T. M. (1997). *Machine learning*. McGraw Hill.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Moerland, T. M., Broekens, J., and Jonker, C. M. (2018). Emotion in reinforcement learning agents and robots: a survey. *Machine Learning*, 107(2):443–480.

Mohamed, S. and Rezende, D. J. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 2125–2133.

Moturu, S. et al. (2011). Using social sensing to understand the links between sleep, mood, and sociability. In *Int. Conf. on Social Comput.*, pages 208–214. IEEE.

Mubaris, N. (2017). Support vector machines for classification.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062.

Murphy, K. (2012). *Mach. learning: a probabilistic perspective*. MIT press.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017). Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785.

Nautiyal, D. (2019). Underfitting and overfitting in machine learning.

Nosakhare, E. and Picard, R. (2019). Probabilistic latent variable modeling for assessing behavioral influences on well-being. In

Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2718–2726. ACM.

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.

Oudeyer, P.-Y. and Kaplan, F. (2006). Discovering communication. *Connection Science*, 18(2):189–206.

Oudeyer, P.-Y., Kaplan, F., et al. (2008). How can we define intrinsic motivation. In *Proc. of the 8th Conf. on Epigenetic Robotics*, volume 5, pages 29–31.

Oudeyer, P.-Y. and Smith, L. B. (2016). How evolution may work through curiosity-driven developmental process. *Topics in Cognitive Science*, 8(2):492–502.

Park, Y., Cho, J., and Kim, G. (2018). A hierarchical latent structure for variational conversation modeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1792–1801.

Partonen, T. (1996). Dopamine and circadian rhythms in seasonal affective disorder. *Medical hypotheses*, 47(3):191–192.

Patel, M., Asch, D., and Volpp, K. (2015). Wearable devices as facilitators, not drivers, of health behavior change. *Jama*, 313(5):459–460.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544.

Pearl, J. (2013). Structural counterfactuals: A brief introduction. *Cognitive science*, 37(6):977–985.

Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal inference in statistics: a primer*. John Wiley & Sons.

Peirce, R. et al. (2000). A longitudinal model of social contact, social support, depression, and alcohol use. *Health Psychology*, 19(1):28.

Pentland, A. (2004). Social dynamics: Signals and behavior. In *Int. Conf. on Developmental Learning*, volume 5.

Perolat, J., Leibo, J. Z., Zambaldi, V., Beattie, C., Tuyls, K., and Graepel, T. (2017). A multi-agent reinforcement learning model of common-pool resource appropriation. In *Advances in Neural Information Processing Systems*, pages 3643–3652.

Peters, J., Mülling, K., and Altun, Y. (2010). Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta.

Peysakhovich, A. and Lerer, A. (2017). Consequentialist conditional cooperation in social dilemmas with imperfect information. *arXiv preprint arXiv:1710.06975*.

Peysakhovich, A. and Lerer, A. (2018). Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2043–2044. International Foundation for Autonomous Agents and Multiagent Systems.

Picard, R. W. (2000). *Affective computing*. MIT press.

Poh, M. et al. (2010). A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *Biomedical Eng.*, 57(5):1243–1252.

Poh, M.-Z. (2011). *Continuous assessment of epileptic seizures with wrist-worn biosensors*. PhD thesis, MIT.

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

Provine, R. R. (2001). *Laughter: A scientific investigation*. Penguin.

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S., and Botvinick, M. (2018). Machine theory of mind. *arXiv preprint arXiv:1802.07740*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Ramirez, R. and Vamvakousis, Z. (2012). Detecting emotion from eeg signals using the emotive epoc device. In *International Conference on Brain Informatics*, pages 175–184. Springer.

Rashkin, H., Smith, E. M., Li, M., and Boureau, Y.-L. (2018). I know the feeling: Learning to converse with empathy. *arXiv preprint arXiv:1811.00207*.

Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, MA.

Ratey, J. (2008). *Spark: The revolutionary new science of exercise and the brain*. Hachette Digital, Inc.

Rawlik, K., Toussaint, M., and Vijayakumar, S. (2012). On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics: science and systems*.

Reinhardt, T. et al. (2012). Salivary cortisol, heart rate, electrodermal activity and subjective stress responses to the mannheim multicomponent stress test (mmst). *Psychiatry research*, 198(1):106–111.

Reis, H. and Gable, S. (2003). Toward a positive psychology of relationships.

Rendell, L., Boyd, R., Cownden, D., Enquist, M., Eriksson, K., Feldman, M. W., Fogarty, L., Ghirlanda, S., Lillicrap, T., and Laland, K. N. (2010). Why copy others? insights from the social learning strategies tournament. *Science*, 328(5975):208–213.

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.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Rokach, L. (2009). *Pattern classification using ensemble methods*. World Scientific.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., et al. (2019). Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*.

Roose, K. (2019). YoutubeâĂŹs product chief on online radicalization and algorithmic rabbit holes. *The New York Times*.

Rosenstein, M. et al. (2005). To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, volume 898.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

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.

Salakhutdinov, R., Tenenbaum, J., and Torralba, A. (2013). Learning with hierarchical-deep models. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1958–1971.

Sanchez, B. (2018). Predicting iris flower species with k-means clustering in python.

Sano, A. (2015). *Measuring College Students' Sleep, Stress and Mental Health with Wearable Sensors and Mobile Phones*. PhD thesis, MIT.

Sano, A. et al. (2015a). Prediction of happy-sad mood from daily behaviors and previous sleep history. In *EMBC*. IEEE.

Sano, A. et al. (2015b). Recognizing academic performance, sleep quality, stress level, and mental health using personality traits, wearable sensors and mobile phones. In *Body Sensor Networks*.

Sano, A. and Picard, R. (2013a). Recognition of sleep dependent memory consolidation with multi-modal sensor data. In *Body Sensor Networks (BSN)*, pages 1–4. IEEE.

Sano, A. and Picard, R. (2013b). Stress recognition using wearable sensors and mobile phones. In *ACII*, pages 671–676. IEEE.

Sano, A. and Picard, R. (2014). Comparison of sleep-wake classification using electroencephalogram and wrist-worn multi-modal sensor data. In *EMBC*, pages 930–933. IEEE.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Schelling, T. C. (1973). Hockey helmets, concealed weapons, and daylight saving: A study of binary choices with externalities. *Journal of Conflict resolution*, 17(3):381–428.

Scherer, S., Morency, L.-P., Gratch, J., and Pestian, J. (2015). Reduced vowel space is a robust indicator of psychological distress: A cross-corpus analysis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4789–4793. IEEE.

Schmidhuber, J. (1991). Adaptive confidence and adaptive curiosity. In *Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2*. Citeseer.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897.

Schultz, W. (2007). Behavioral dopamine signals. *Trends in neurosciences*, 30(5):203–210.

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. (2017). Generating focussed molecule libraries for drug discovery with recurrent neural networks. *arXiv preprint arXiv:1701.01329*.

Seligman, M. (2012). *Flourish: A visionary new understanding of happiness and well-being*. Simon and Schuster.

Senechal, T., McDuff, D., and Kaliouby, R. (2015). Facial action unit detection using active learning and an efficient non-linear kernel approximation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18.

Sequeira, P., Melo, F. S., Prada, R., and Paiva, A. (2011). Emerging social awareness: Exploring intrinsic motivation in multiagent learning. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–6. IEEE.

Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., et al. (2017a). A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., and Bengio, Y. (2017b). A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Shah, P., Hakkani-Tur, D., Liu, B., and Tur, G. (2018). Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Confer-*

ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pages 41–51.

Shcherbakov, O. and Batishcheva, V. (2014). Image inpainting based on stacked autoencoders. In *Journal of Physics: Conference Series*, volume 536, page 012020. IOP Publishing.

Shen, X., Su, H., Niu, S., and Demberg, V. (2018). Improving variational encoder-decoders in dialogue generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shi, W. and Yu, Z. (2018). Sentiment adaptive end-to-end dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1509–1519.

Shin, J., Xu, P., Madotto, A., and Fung, P. (2019). Happybot: Generating empathetic dialogue responses by improving user experience lookahead. *arXiv preprint arXiv:1906.08487*.

Shivhare, S. N. and Khethawat, S. (2012). Emotion detection from text. *arXiv preprint arXiv:1205.4944*.

Shum, M., Kleiman-Weiner, M., Littman, M. L., and Tenenbaum, J. B. (2019). Theory of minds: Understanding behavior in groups through inverse planning. *arXiv preprint arXiv:1901.06085*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Singh, S. P., Barto, A. G., and Chentanez, N. (2004). Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1281–1288.

Srivastava, N. et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Stavropoulos, K. K. and Carver, L. J. (2013). Research review: social motivation and oxytocin in autism–implications for joint attention development and intervention. *Journal of Child Psychology and Psychiatry*, 54(6):603–618.

Stengel, R. F. (1986). *Stochastic optimal control*. John Wiley and Sons New York, New York.

Still, S. and Precup, D. (2012). An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131(3):139–148.

Storm, H. et al. (2000). The development of a software program for analyzing spontaneous and externally elicited skin conductance changes in infants and adults. *Clin. Neurophysiology*, 111(10):1889–1898.

Strouse, D., Kleiman-Weiner, M., Tenenbaum, J., Botvinick, M., and Schwab, D. (2018). Learning to share and hide intentions using information regularization. *arXiv preprint arXiv:1808.02093*.

Su, P.-H., Budzianowski, P., Ultes, S., Gasic, M., and Young, S. (2017). Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157.

Suhara, Y., Xu, Y., and Pentland, A. (2017). Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 715–724. International World Wide Web Conferences Steering Committee.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147.

Sutton and Barto (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tajima, S., Yanagawa, T., Fujii, N., and Toyoizumi, T. (2015). Untangling brain-wide dynamics in consciousness by cross-embedding. *PLoS computational biology*, 11(11):e1004537.

Taylor, S., Jaques, N., Chen, W., Fedor, S., Sano, A., and Picard, R. (2015). Automatic identification of artifacts in electrodermal activity data. In *EMBC*. IEEE.

Taylor, S. A., Jaques, N., Nosakhare, E., Sano, A., and Picard, R. (2017). Personalized multitask learning for predicting tomorrow's mood, stress, and health. *IEEE Transactions on Affective Computing*.

Thomas, P. and Brunskill, E. (2016). Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148.

Thomas, P., Theocharous, G., and Ghavamzadeh, M. (2015). High confidence policy improvement. In *International Conference on Machine Learning*, pages 2380–2388.

Thomaz, A. L. and Breazeal, C. (2007). Robot learning via socially guided exploration. *Development and Learning*, pages 82–87.

Thomaz, A. L. and Cakmak, M. (2013). Active social learning in humans and robots. *Social learning theory: Phylogenetic considerations across animal, plant, and microbial taxa, ed. KB Clark*, pages 113–28.

Tian, Y.-I., Kanade, T., and Cohn, J. F. (2001). Recognizing action units for facial expression analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 23(2):97–115.

Todorov, E. (2007). Linearly-solvable markov decision problems. In *Advances in neural information processing systems (NIPS)*, pages 1369–1376.

Tomasello, M. (2009a). *The cultural origins of human cognition*. Harvard university press.

Tomasello, M. (2009b). *Why we cooperate*. MIT press.

Truong, K. et al. (2014). Slide to x: unlocking the potential of smartphone unlocking. In *Human factors in comput. systems*, pages 3635–3644. ACM.

Tsuno, N. et al. (2005). Sleep and depression. *J. of Clin. Psychiatry*.

Umematsu, T., Sano, A., and Picard, R. (2019). Daytime data and lstm can forecast tomorrow's stress, health, and happiness. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*.

Valenza, G., Citi, L., Lanatá, A., Scilingo, E. P., and Barbieri, R. (2014). Revealing real-time emotional responses: a personalized assessment based on heartbeat dynamics. *Scientific reports*, 4:4998.

Valstar, M. et al. (2012). Meta-analysis of the first facial expression recognition challenge. *Systems, Man, and Cybernetics*, 42(4):966–979.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

van Schaik, C. P. and Burkart, J. M. (2011). Social learning and evolution: the cultural intelligence hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1567):1008–1016.

van Vugt, H., Hoorn, J., and Konijn, E. (2009). Interactive engagement with embodied agents: An empirically validated framework. *Comp. Animation and Virtual Worlds*, 20(2-3):195–204.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Veenhoven, R. (2008). Healthy happiness: Effects of happiness on physical health and the consequences for preventive health care. *Journal of happiness studies*, 9(3):449–469.

Venkatesh, A., Khatri, C., Ram, A., Guo, F., Gabriel, R., Nagar, A., Prasad, R., Cheng, M., Hedayatnia, B., Metallinou, A., et al. (2018). On evaluating and comparing conversational agents. *arXiv preprint arXiv:1801.03625*, 4:60–68.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408.

von Frisch, K. (1969). The dance language and orientation of bees. 5.

Vyzas, E. (1999). *Recognition of emotional and cognitive states using physiological data*. PhD thesis, MIT.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732.

Weininger, D. (1970). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. In *Proc. Edinburgh Math. SOC*, volume 17, pages 1–14.

Westefeld, J. and Furr, S. (1987). Suicide and depression among college students. *Professional Psychology: Research and Practice*, 18(2):119.

Wilhelm, F. H. and Roth, W. T. (1998). Taking the laboratory to the skies: Ambulatory assessment of self-report, autonomic, and respiratory responses in flying phobia. *Psychophysiology*, 35(5):596–606.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Wilson, A. et al. (2007). Multi-task reinforcement learning: a hierarchical bayesian approach. In *ICML*, pages 1015–1022. ACM.

Wilson, D. R. and Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural networks*, 16(10):1429–1451.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390.

Wong, J. and McGee, K. (2012). Frown more, talk more: effects of facial expressions in establishing conversational rapport with virtual agents. In *IVA*, pages 419–425.

Xia, R. and Liu, Y. (2015). A multi-task learning framework for emotion recognition using 2d continuous space. *IEEE Transactions on Affective Computing*.

Xie, J., Xu, L., and Chen, E. (2012). Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349.

Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., and Ma, W.-Y. (2017). Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Xu, Y. et al. (1994). Wavelet transform domain filters: a spatially selective noise filtration technique. *Image Processing*, 3(6):747–758.

Xue, W., Huang, Z., Luo, X., and Mao, Q. (2015). Learning speech emotion features by joint disentangling-discrimination. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pages 374–379. IEEE.

Xue, Y. et al. (2007). Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:35–63.

Yang, Z., Metallinou, A., and Narayanan, S. (2014). Analysis and predictive modeling of body language behavior in dyadic interactions from multimodal interlocutor cues. *Multimedia*, 16(6):1766–1778.

Yu, C., Zhang, M., and Ren, F. (2013). Emotional multiagent reinforcement learning in social dilemmas. In *International Conference* *on Principles and Practice of Multi-Agent Systems*, pages 372–387. Springer.

Yu, N. (2007). Gradient methods for minimizing composite objective function. *CORE Discussion Papers*.

Zangróniz, R., Martínez-Rodrigo, A., Pastor, J., López, M., and Fernández-Caballero, A. (2017). Electrodermal activity sensor for classification of calm/distress condition. *Sensors*, 17(10):2324.

Zhang, B., Provost, E. M., and Essl, G. (2017). Cross-corpus acoustic emotion recognition with multi-task learning: Seeking common ground while preserving differences. *IEEE Transactions on Affective Computing*.

Zhang, C. and Zhang, Z. (2014). Improving multiview face detection with multi-task deep convolutional neural networks. In *Applications of Computer Vision*, pages 1036–1041. IEEE.

Zhao, Papangelis, and Cassell (2014). Towards a dyadic computational model of rapport management for human-virtual agent interaction. In *Intelligent Virtual Agents*, pages 514–527.

Zhao, T., Zhao, R., and Eskenazi, M. (2017). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664.

Zhou, H., Huang, M., Zhang, T., Zhu, X., and Liu, B. (2018a). Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhou, L., Gao, J., Li, D., and Shum, H.-Y. (2018b). The design and implementation of xiaoice, an empathetic social chatbot. *arXiv preprint arXiv:1812.08989*.

Zhou, X. and Wang, W. Y. (2018). Mojitalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1128–1137.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

3 *Multi-agent social learning via causal influence*

3.1 Introduction 102
3.2 Sequential Social Dilemmas 104
3.3 Multi-agent RL for SSDs 106
3.4 Basic social influence 107
3.4.1 Experiment I: Basic influence 109
3.5 Influential communication 111
3.5.1 Experiment II: Influential communication 112
3.6 Modeling other agents 115
3.6.1 Experiment III: Modeling other agents 117
3.7 Related work 117
3.8 Details on causal inference 120
3.9 Conclusions and future work 121
3.10 Statement of contributions 122
3.11 Appendix 122
3.11.1 Influence as Mutual Information 122
3.11.2 Additional experiment - Box Trapped 124
3.11.3 Implementation details 125
3.11.4 Additional results 128

Social learning is incredibly important for humans, and has been linked to our ability to achieve unprecedented progress and coordination on a massive scale (Henrich, 2015; Harari, 2014; Laland, 2017; van Schaik and Burkart, 2011; Herrmann et al., 2007). In this chapter, we propose a method for allowing reinforcement learning (RL) agents in a multi-agent environment to learn socially from the actions taken by other agents. Specifically, we propose a unified mechanism for achieving coordination and communication in Multi-Agent Reinforcement Learning (MARL) through rewarding agents for having causal influence over other agents' actions. Empirical results demonstrate that influence leads to enhanced coordination and communication in challenging social dilemma environments, dramatically increasing the learning curves of the deep RL agents, and leading to more meaningful learned communication protocols. The influence rewards for all agents can be computed in a decentralized way by enabling agents to learn a model of other agents using deep neural networks. In contrast, key previous works on emergent communication in the MARL setting were unable to learn diverse policies in a decentralized manner and had to resort to centralized training. Consequently, the influence reward opens up a window of new opportunities for research in this area.

3.1 Introduction

Intrinsic Motivation for Reinforcement Learning (RL) refers to reward functions that allow agents to learn useful behavior across a variety of tasks and environments, sometimes in the absence of environmental reward (Singh et al., 2004). Previous approaches to intrinsic motivation often focus on curiosity (e.g. Pathak et al. (2017); Schmidhuber (2010)), or empowerment (e.g. Klyubin et al. (2005); Mohamed and Rezende (2015)). Here, we consider the problem of deriving intrinsic social motivation from other agents. While some previous work has investigated intrinsic social motivation for RL (e.g. Sequeira et al. (2011); Hughes et al. (2018); Peysakhovich and Lerer (2018)), these approaches rely on hand-crafted rewards specific to the environment, or allowing agents to view the rewards obtained by other agents. Such assumptions make it impossible to achieve independent training of MARL agents across multiple environments.

Achieving coordination among agents in MARL still remains a difficult problem. Prior work in this domain (e.g., Foerster et al. (2017, 2016)), often resorts to centralized training to ensure that agents learn to coordinate. While communication among agents could help with coordination, training emergent communication protocols also remains a challenging problem; recent empirical results underscore the difficulty of learning meaningful emergent communication protocols, even when relying on centralized training (e.g., Lazaridou et al. (2018); Cao et al. (2018); Foerster et al. (2016)).

We propose a unified method for achieving both coordination and communication in MARL by giving agents an intrinsic reward for having a causal influence on other agents' actions. Causal influence is assessed using counterfactual reasoning; at each timestep, an agent simulates alternate, counterfactual actions that it could have taken, and assesses their effect on another agent's behavior. Actions that lead to relatively higher change in the other agent's behavior are considered to be highly influential and are rewarded. We show how this reward is related to maximizing the mutual information between agents' actions, and hypothesize that this inductive bias will drive agents to learn coordinated behavior. Maximizing mutual information as a form of intrinsic motivation has been studied in the literature on empowerment (e.g. Klyubin et al. (2005); Mohamed and Rezende (2015)). Social influence can be seen as a novel, social form of empowerment.

To study our influence reward, we adopt the Sequential Social Dilemma (SSD) multi-agent environments of Leibo et al. (2017). Through a series of three experiments, we show that the proposed social influence reward allows agents to learn to coordinate and communicate more effectively in these SSDs. We train recurrent neural network policies directly from pixels, and show in the first experiment that deep RL agents trained with the proposed social influence reward learn effectively and attain higher collective reward than powerful baseline deep RL agents, which often completely fail to learn.

In the second experiment, the influence reward is used to directly train agents to use an explicit communication channel. We demonstrate that the communication protocols trained with the influence reward are more meaningful and effective for obtaining better collective outcomes. Further, we find a significant correlation between being influenced through communication messages and obtaining higher individual reward, suggesting that influential communication is beneficial to the agents that receive it. By examining the learning curves in this second experiment, we again find that the influence reward is essential to allow agents to learn to coordinate.

Finally, we show that influence agents can be trained independently, when each agent is equipped with an internal neural network *Model of Other Agents* (MOA), which has been trained to predict the actions of every other agent. The agent can then simulate counterfactual actions and use its own internal MOA to predict how these will affect other agents, thereby computing its own intrinsic influence reward. Influence agents can thus learn socially, just by observing other agents' actions, and without requiring a centralized controller or access to another agent's reward function. Therefore, the influence reward offers us a simple, general and effective way of overcoming long-standing unrealistic assumptions and limitations in this field of research, including centralized training and the sharing of reward functions or policy parameters. Moreover, both the influence rewards as well as the agents' policies can be learned directly from pixels using expressive deep recurrent neural networks. In this third experiment, the learning curves once again show that the influence reward dramatically enhances coordination in these complex domains.

The chapter is structured as follows. We describe the environments in Section 3.2, and the MARL setting in Section 3.3. Section 3.4 introduces the basic formulation of the influence reward, Section 3.5 extends it with the inclusion of explicit communication protocols, and Section 3.6 advances it by including models of other agents to achieve independent training. Each of these three sections presents experiments and results that empirically demonstrate the efficacy of the social influence reward. Related work is presented in Section 3.7. Finally, more details about the causal inference procedure are given in Section 3.8.

3.2 *Sequential Social Dilemmas*

Sequential Social Dilemmas (SSDs) (Leibo et al., 2017) are partially observable, spatially and temporally extended multi-agent games with a game-theoretic payoff structure. An individual agent can obtain higher reward in the short-term by engaging in defecting, non-cooperative behavior (and thus is greedily motivated to defect), but the total payoff per agent will be higher if all agents cooperate. Thus, the collective reward obtained by a group of agents in these SSDs gives a clear signal about how well the agents learned to cooperate (Hughes et al., 2018).

We experiment with two SSDs in this work, a public goods game *Cleanup*, and a public pool resource game *Harvest* (see Figure 3.1). Both games are partially observable, and the rewards are based on collecting apples[1], which are a limited resource. In *Cleanup* (a public goods game) agents must clean a river before apples can grow, but are not able to harvest apples while cleaning. In *Harvest* (a common pool resource game), apples respawn at a rate proportional to the amount of nearby apples; if apples are harvested too quickly, they will not grow back. Both coordination, and cooperation are required to solve both games. In *Cleanup*, agents must

[1] Apples are worth a reward of $+1$. Agents have the ability to punish each other with a *fining beam*, which costs -1 reward to fire, and fines any agent it hits -50 reward.

efficiently time harvesting apples and cleaning the river, and allow agents cleaning the river a chance to consume apples. In *Harvest*, agents must spatially distribute their harvesting, and abstain from consuming apples too quickly in order to harvest sustainably. Agents must coordinate harvesting apples with the behavior of other agents in order to achieve cooperation. For reproducibility, the code for these games has been made available in open-source: `https://github.com/eugenevinitsky/sequential_social_dilemma_games`.

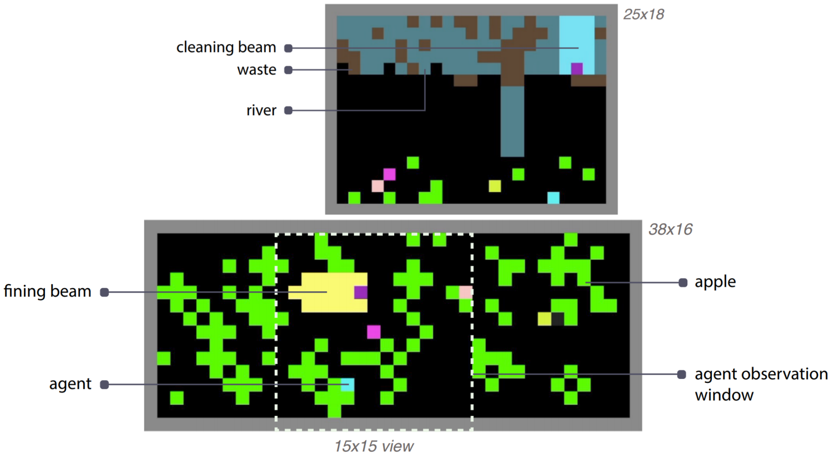
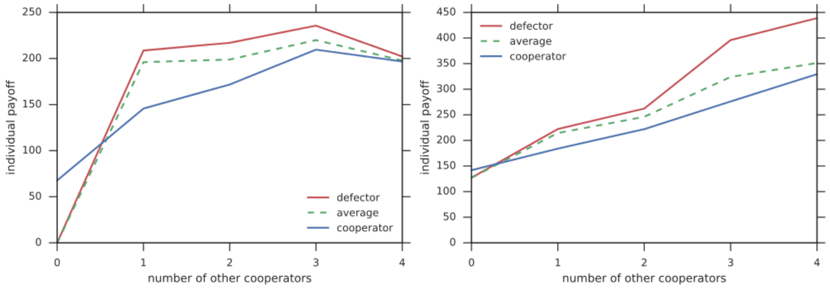


Figure 3.1: The two SSD environments, *Cleanup* (top) and *Harvest* (bottom). Agents can exploit other agents for immediate payoff, but at the expense of the long-term collective reward of the group. Reproduced with permission from Hughes et al. (2018).

The reward structure of the games is shown in Figure 3.2, which gives the Schelling diagram for both SSD tasks under investigation. A Schelling diagram (Schelling, 1973; Perolat et al., 2017) depicts the relative payoffs for a single agent's strategy given the number of other agents who are cooperative out of a fixed total number of other agents (in this case, there are always five agents). As the Schelling diagrams in Figure 3.2 reveal, all agents would benefit from learning to cooperate in these games, because even agents that are being exploited get higher reward than in the regime where more agents defect. However, traditional RL agents struggle to learn to coordinate or cooperate to solve these tasks effectively (Hughes et al., 2018). Thus, these SSDs represent challenging benchmark tasks for the social influence reward. Not only must influence agents learn to coordinate their behavior to obtain high reward, they must also learn to cooperate.



(a) *Cleanup*

(b) *Harvest*

Figure 3.2: Schelling diagrams for the two social dilemma tasks show that an individual agent is motivated to defect, though everyone benefits when more agents cooperate. Reproduced with permission from Hughes et al. (2018).

3.3 *Multi-agent RL for SSDs*

We consider a MARL Markov game defined by the tuple $\langle S, T, A, r \rangle$, in which multiple agents are trained to independently maximize their own individual reward; agents do not share weights. The environment state is given by $s \in \mathcal{S}$. At each timestep t, each agent k chooses an action $a_t^k \in \mathcal{A}$. The actions of all N agents are combined to form a joint action $\boldsymbol{a}_t = [a_t^0, ...a_t^N]$, which produces a transition in the environment $T(s_{t+1}|\boldsymbol{a}_t, s_t)$, according to the state transition distribution T. Each agent then receives its own reward $r^k(\boldsymbol{a}_t, s_t)$, which may depend on the actions of other agents. A history of these variables over time is termed a trajectory, $\tau = \{s_t, \boldsymbol{a}_t, \boldsymbol{r}_t\}_{t=0}^{T}$. We consider a partially observable setting in which the kth agent can only view a portion of the true state, s_t^k. Each agent seeks to maximize its own total expected discounted future reward, $R^k = \sum_{i=0}^{\infty} \gamma^i r_{t+i}^k$, where γ is the discount factor. A distributed asynchronous advantage actor-critic (A3C) approach (Mnih et al., 2016) is used to train each agent's policy π^k.

Our neural networks consist of a convolutional layer, fully connected layers, a Long Short Term Memory (LSTM) recurrent layer (Gers et al., 1999), and linear layers. All networks take images as input and output both the policy π^k and the value function $V^{\pi_k}(s)$, but some network variants consume additional inputs and output either communication policies or models of other agents' behavior. We will refer to the internal LSTM state of the kth agent at timestep t as u_t^k.

3.4 *Basic social influence*

Social influence intrinsic motivation gives an agent additional reward for having a causal influence on another agent's actions. Specifically, it modifies an agent's immediate reward so that it is defined as $r_t^k = \alpha e_t^k + \beta c_t^k$, where e_t^k is the extrinsic or environmental reward, and c_t^k is the causal influence reward.

To compute the causal influence of one agent on another, suppose there are two agents, *k* and *j*, and that agent *j* is able to condition its policy on agent *k*'s action at time *t*, a_t^k. Thus, agent *j* computes the probability of its next action as $p(a_t^j|a_t^k, s_t^j)$. We can then intervene on a_t^k by replacing it with a counterfactual action, $\tilde{a}_t^k$. This counterfactual action is used to compute a new distribution over *j*'s next action, $p(a_t^j|\tilde{a}_t^k, s_t^j)$. Essentially, agent *k* asks a retrospective question: "How would *j*'s action change if I had acted differently in this situation?".

By sampling several counterfactual actions, and averaging the resulting policy distribution of *j* in each case, we obtain the marginal policy of *j*, $p(a_t^j|s_t^j) = \sum_{\tilde{a}_t^k} p(a_t^j|\tilde{a}_t^k, s_t^j)p(\tilde{a}_t^k|s_t^j)$; in other words, *j*'s policy if it did not consider agent *k*. The discrepancy between the marginal policy of *j* and the conditional policy of *j* given *k*'s action is a measure of the causal influence of *k* on *j*; it gives the degree to which *j* changes its planned action distribution because of *k*'s action. Thus, the causal influence reward for agent *k* is defined as:

$$\begin{aligned} c_t^k &= \sum_{j=0, j\neq k}^{N} \Big[D_{KL}[p(a_t^j \mid a_t^k, s_t^j) \Big\| \sum_{\tilde{a}_t^k} p(a_t^j \mid \tilde{a}_t^k, s_t^j) p(\tilde{a}_t^k \mid s_t^j)] \Big] \\ &= \sum_{j=0, j\neq k}^{N} \Big[D_{KL}[p(a_t^j \mid a_t^k, s_t^j) \Big\| p(a_t^j \mid s_t^j)] \Big]. \end{aligned} \tag{3.1}$$

Note that it is possible to use a divergence metric other than KL; we have found empirically that the influence reward is robust to the choice of metric.

The reward in Eq. 3.1 is related to the mutual information (MI) between the actions of agents *k* and *j*, $I(a^k; a^j|s)$. As the reward is computed over many trajectories sampled independently from the environment, we obtain a Monte-Carlo estimate of $I(a^k; a^j|s)$. In expectation, the influence reward incentivizes agents to maximize the mutual information between their actions. The proof is given in Section 3.11.1 of the Supplementary Material. Intuitively, training agents to maximize the MI between their actions results in more coordinated behavior.

For the basic influence model presented in this section we make two assumptions: 1) we use centralized training to compute c_t^k directly from the policy of agent j, and 2) we assume that influence is unidirectional: agents trained with the influence reward can only influence agents that are not trained with the influence reward (the sets of influencers and influencees are disjoint, and the number of influencers is in $[1, N-1]$). Algorithm 2 shows how to compute the influence reward in this simplified case. In later sections, we relax both of these assumptions. In particular, Section 3.6 gives an alternative algorithm that allows each agent to compute its own influence reward internally, without viewing any information about the other agents beyond their actions. Further explanations of the causal inference procedure (including causal diagrams) are available in Section 3.8.

Algorithm 2: Computing basic influence reward for agent k

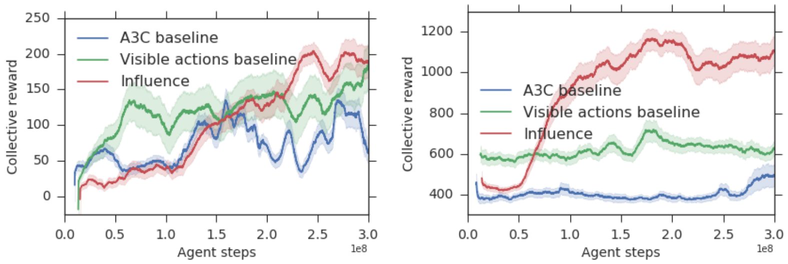
```
Require: $\tau$, trajectory containing actions and observations for all
agents. Let $T$ be the trajectory length.
influence $= [0] * T$
for timestep $t$ in $[0, T)$ do
    for other agent $j$ in $N$ do
        prob_aj $= 0$
        for action $\tilde{a}_t^k$ in $\mathcal{A}$ do
            Compute $p(\tilde{a}_t^k | s_t^k; \theta^k)$ using agent $k$'s policy network
            Compute $p(a_t^j | \tilde{a}_t^k, s_t^j; \theta^j)$ using agent $j$'s policy network
            prob_aj $=$ prob_aj $+ p(a_t^j | \tilde{a}_t^k, s_t^j) p(\tilde{a}_t^k | s_t^k)$
        end
        $p(a_t^j | s_t^j) \leftarrow$ prob_aj
        influence$[t] =$ influence$[t] + D_{KL}[p(a_t^j | a_t^k, s_t^j) || p(a_t^j | s_t^j)]$
    end
end
```

The variance of policy gradient updates increases as the number of agents in the environment grows (Lowe et al., 2017b). This issue can hinder convergence to equilibrium for large-scale MARL tasks. Social influence can reduce the variance of policy gradients by introducing explicit dependencies across the actions of each agent. This is because the conditional variance of the gradients an agent is receiving will be less than or equal to the marginalized variance.

3.4.1 *Experiment I: Basic influence*

Figure 3.3 shows the results of testing agents trained with the basic influence reward against standard A3C agents, and an ablated version of the model in which agents do not receive the influence reward, but are able to condition their policy on the actions of other agents (even when the other agents are not within the agent's partially observed view of the environment). We term this ablated model the *visible actions baseline*. In this and all other results figures, we measure the total collective reward obtained using the best hyperparameter setting tested with 5 random seeds each. Error bars show a 99.5% confidence interval (CI) over the random seeds, computed within a sliding window of 200 agent steps. We use a curriculum learning approach which gradually increases the weight of the social influence reward over C steps ($C \in [0.2 - 3.5] \times 10^8$); this sometimes leads to a slight delay before the influence models' performance improves.

As is evident in Figure 3.3, introducing an awareness of other agents' actions helps, but having the social influence reward eventually leads to significantly higher collective reward in both games. Due to the structure of the SSD games, we can infer that agents that obtain higher collective reward learned to cooperate more effectively. In the *Harvest* MARL setting, it is clear that the influence reward is essential to achieve any reasonable learning.



(a) *Cleanup* (b) *Harvest*

Figure 3.3: Total collective reward obtained in Experiment 1. Agents trained with influence (red) significantly outperform the baseline and ablated agents. In Harvest, the influence reward is essential to achieve any meaningful learning.

To understand how social influence helps agents achieve cooperative behavior, we investigated the trajectories produced by high scoring models in both *Cleanup* and *Harvest*; the analysis revealed interesting behavior. As an example, in the *Cleanup* video available here: `https://youtu.be/iH_V5WKQxmo` a single agent (shown in purple) was trained with the social influence reward. Unlike the other agents, which continue to move and explore randomly while waiting for apples to spawn, the influencer only traverses the map when it is pursuing an apple, then stops. The rest of the time it stays still.

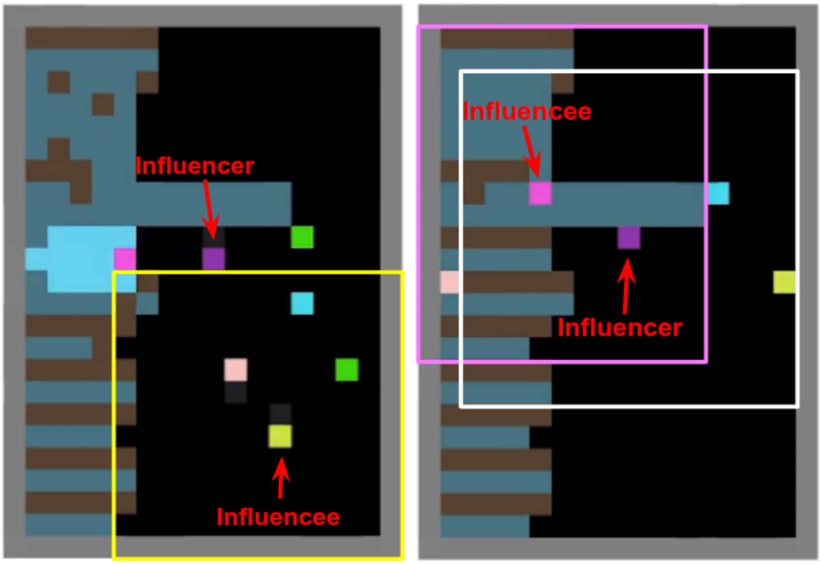


Figure 3.4: Moments of high influence. Left: the purple influencer gains influence by moving, signaling the presence of an apple (green tiles) outside the yellow influencee's field-of-view (yellow outlined box). Right: the purple influencer stays still, signaling to the pink influencee that no apples have yet appeared.

Figure 3.4 shows a moment of high influence between the influencer and the yellow influencee. The influencer has chosen to move towards an apple that is outside of the ego-centric field-of-view of the yellow agent. Because the influencer only moves when apples are available, this signals to the yellow agent that an apple must be present above it which it cannot see. This changes the yellow agent's distribution over its planned action, $p(a_t^j | a_t^k, s_t^j)$, and allows the purple agent to gain influence. A similar moment occurs when the influencer signals to an agent that has been cleaning the river that no apples have appeared by staying still (Figure 3.4 right).

In this case study, the influencer agent learned to use its own actions as a binary code which signals the presence or absence of apples in the environment. We observe a similar effect in *Harvest*. This type of action-based communication could be likened to the bee waggle dance discovered by von Frisch (1969). Evidently, the influence reward gave rise not only to cooperative behavior, but to emergent communication.

It is important to consider the limitations of the influence reward. Whether it will always give rise to cooperative behavior may depend on the specifics of the environment and task, and tuning the trade-off between environmental and influence reward. Although influence is arguably necessary for coordination (e.g. two agents coordinating to manipulate an object must have a high degree of influence between their actions), it may be possible to influence another agent in a non-cooperative way. The results provided here show that the influence reward did lead to increased cooperation, in spite of cooperation being difficult to achieve in these environments.

3.5 *Influential communication*

Given the above results, we next experiment with using the influence reward to train agents to use an explicit communication channel. We take some inspiration from research drawing a connection between influence and communication in human learning. According to Melis and Semmann (2010), human children rapidly learn to use communication to influence the behavior of others when engaging in cooperative activities. They explain that "this ability to influence the partner via communication has been interpreted as evidence for a capacity to form shared goals with others", and that this capacity may be "what allows humans to engage in a wide range of cooperative activities".

Thus, we equip agents with an explicit communication channel, similar to the approach used by Foerster et al. (2016). At each timestep, each agent k chooses a discrete communication symbol m_t^k; these symbols are concatenated into a combined message vector $\boldsymbol{m}_t = [m_t^0, m_t^1 ... m_t^N]$, for N agents. This message vector $\boldsymbol{m}_t$ is then given as input to every other agent in the next timestep. Note that previous work has shown that self-interested agents do not learn to use this type of ungrounded, *cheap talk* communication channel effectively (Crawford and Sobel, 1982; Cao et al., 2018; Foerster et al., 2016; Lazaridou et al., 2018).

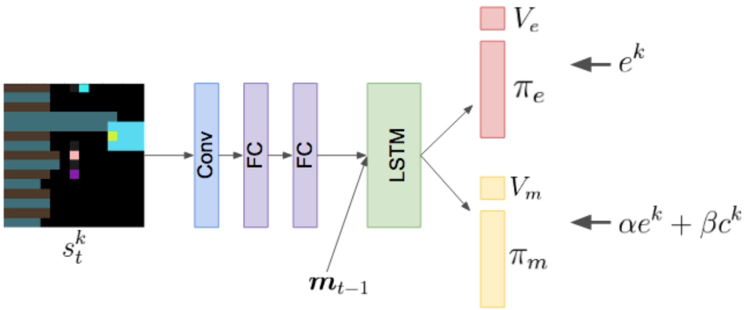


Figure 3.5: The communication model has two heads, which learn the environment policy, π_e, and a policy for emitting communication symbols, π_m. Other agents' communication messages $\boldsymbol{m}_{t-1}$ are input to the LSTM.

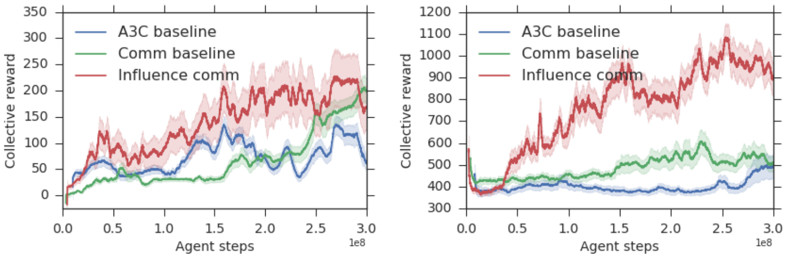
To train the agents to communicate, we augment our initial network with an additional A3C output head, that learns a communication policy π_m and value function V_m to determine which symbol to emit (see Figure 3.5). The normal policy and value function used for acting in the environment, π_e and V_e, are trained only with environmental reward e. We use the influence reward as an additional incentive for training the communication policy, π_m, such that $r = \alpha e + \beta c$. Counterfactuals are employed to assess how much influence an agent's communication message from the previous timestep, m^k_{t-1}, has on another agent's action, a^j_t, where:

$$c^k_t = \sum_{j=0, j \neq k}^{N} \Big[D_{KL}[p(a^j_t \mid m^k_{t-1}, s^j_t) \Big\| p(a^j_t \mid s^j_t)] \Big] \tag{3.2}$$

Importantly, rewarding influence through a communication channel does not suffer from the limitation mentioned in the previous section, i.e. that it may be possible to influence another agent in a non-cooperative way. We can see this for two reasons. First, there is nothing that compels agent j to act based on agent k's communication message; if m^k_t does not contain valuable information, j is free to ignore it. Second, because j's action policy π_e is trained only with environmental reward, j will only change its intended action as a result of observing m^k_t (i.e. be influenced by m^k_t) if it contains information that helps j to obtain environmental reward. Therefore, we hypothesize that influential communication must provide useful information to the listener.

3.5.1 *Experiment II: Influential communication*

Figure 3.6 shows the collective reward obtained when training the agents to use an explicit communication channel. Here, the ablated model has the same structure as in Figure 3.5, but the communication policy π_m is trained only with environmental reward. We observe that the agents incentivized to communicate via the social influence reward learn faster, and achieve significantly higher collective reward for the majority of training in both games. In fact, in the case of *Cleanup*, we found that $\alpha = 0$ in the optimal hyperparameter setting, meaning that it was most effective to train the communication head with zero extrinsic reward (see Table 3.2 in the Supplementary Material). This suggests that influence alone can be a sufficient mechanism for training an effective communication policy. In *Harvest*, once again influence is critical to allow agents to learn coordinated policies and attain high reward.



(a) *Cleanup*

(b) *Harvest*

Figure 3.6: Total collective reward for deep RL agents with communication channels. Once again, the influence reward is essential to improve or achieve any learning.

To analyze the communication behaviour learned by the agents, we introduce three metrics, partially inspired by Bogin et al. (2018). *Speaker consistency*, is a normalized score $\in [0,1]$ which assesses the entropy of $p(a^k|m^k)$ and $p(m^k|a^k)$ to determine how consistently a *speaker* agent emits a particular symbol when it takes a particular action, and vice versa (the formula is given in the Supplementary Material Section 3.11.3.4). We expect this measure to be high if, for example, the speaker always emits the same symbol when it is cleaning the river. We also introduce two measures of *instantaneous coordination* (IC), which are both measures of mutual information (MI): (1) symbol/action IC $= I(m^k_t; a^j_{t+1})$ measures the MI between the influencer/speaker's symbol and the influencee/listener's next action, and (2) action/action IC $= I(a^k_t; a^j_{t+1})$ measures the MI between the influencer's action and the influencee's next action. To compute these measures we first average over all trajectory steps, then take the maximum value between any two agents, to determine if any pair of agents are coordinating. Note that these measures are all *instantaneous*, as they consider only short-term dependencies across two consecutive timesteps, and cannot capture if an agent communicates influential compositional messages, i.e. information that requires several consecutive symbols to transmit and only then affects the other agents behavior.

Figure 3.7 presents the results. The speaker consistencies metric reveals that influence agents more unambiguously communicate about their own actions than baseline agents, indicating that the emergent communication is more meaningful. The IC metrics demonstrate that baseline agents show almost no signs of co-ordinating behavior with communication, i.e. speakers saying A and listeners doing B consistently. This result is aligned with both theoretical results in cheap-talk literature (Crawford and Sobel, 1982), and recent empirical results in MARL (e.g. Foerster et al. (2016); Lazaridou et al. (2018); Cao et al. (2018)).

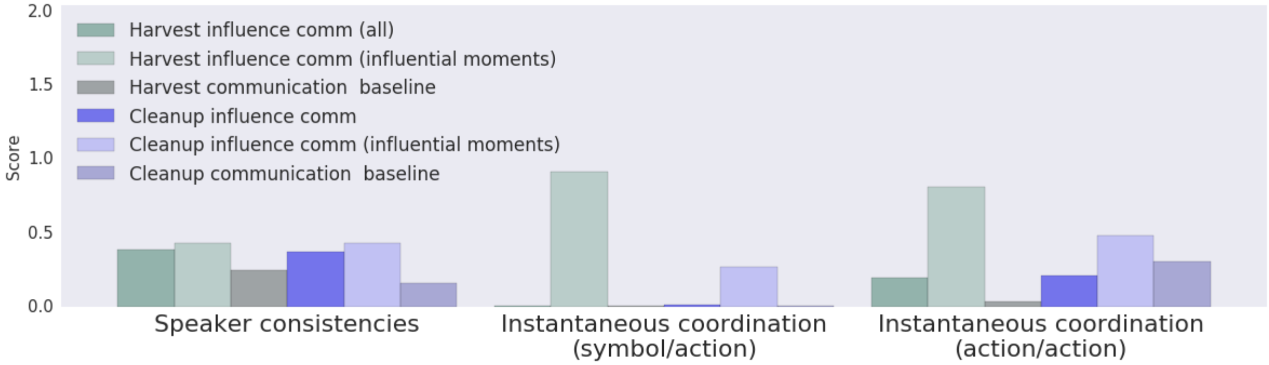


Figure 3.7: Metrics describing the quality of learned communication protocols. The models trained with influence reward exhibit more consistent communication and more coordination, especially in moments where influence is high.

In contrast, we do see high IC between influence agents, but only when we limit the analysis to timesteps on which influence was greater than or equal to the mean influence (cf. *influential moments* in Figure 3.7). Inspecting the results reveals a common pattern: influence is sparse in time. An agent's influence is only greater than its mean influence in less than 10% of timesteps. Because the listener agent is not compelled to listen to any given speaker, listeners selectively listen to a speaker only when it is beneficial, and influence cannot occur all the time. Only when the listener decides to change its action based on the speaker's message does influence occur, and in these moments we observe high $I(m_t^k; a_{t+1}^j)$. It appears the influencers have learned a strategy of communicating meaningful information about their own actions, and gaining influence when this becomes relevant enough for the listener to act on it.

Examining the relationship between the degree to which agents were influenced by communication and the reward they obtained gives a compelling result: agents that are the most influenced also achieve higher individual environmental reward. We sampled 100 different experimental conditions (i.e., hyper-parameters and random seeds) for both games, and normalized and correlated the influence and individual rewards. We found that agents who are more often influenced tend to achieve higher task reward in both *Cleanup*, $\rho = .67$, $p < 0.001$, and *Harvest*, $\rho = .34$, $p < 0.001$. This supports the hypothesis that in order to influence another agent via communication, the communication message should contain information that helps the listener maximize its own environmental reward. Since better listeners/influencees are more successful in terms of task reward, we have evidence that useful information was transmitted to them.

This result is promising, but may depend on the specific experimental approach taken here, in which agents interact with each other repeatedly. In this case, there is no advantage to the speaker for communicating unreliable information (i.e. lying), because it would lose influence with the listener over time. This may not be guaranteed in one-shot interactions. However, given repeated interactions, the above results provide empirical evidence that social influence as intrinsic motivation allows agents to learn meaningful communication protocols when this is otherwise not possible.

3.6 *Modeling other agents*

Computing the causal influence reward as introduced in Section 3.4 requires knowing the probability of another agent's action given a counterfactual, which we previously solved by using a centralized training approach in which agents could access other agents' policy networks. While using a centralized training framework is common in MARL (e.g. Foerster et al. (2017, 2016)), it is less realistic than a scenario in which each agent is trained independently. We can relax this assumption and achieve independent training by equipping each agent with its own internal *Model of Other Agents* (MOA). The MOA consists of a second set of fully-connected and LSTM layers connected to the agent's convolutional layer (see Figure 3.8), and is trained to predict all other agents' next actions given their previous actions, and the agent's egocentric view of the state: $p(\boldsymbol{a}_{t+1}|\boldsymbol{a}_t, s_t^k)$. The MOA is trained using observed action trajectories and cross-entropy loss.

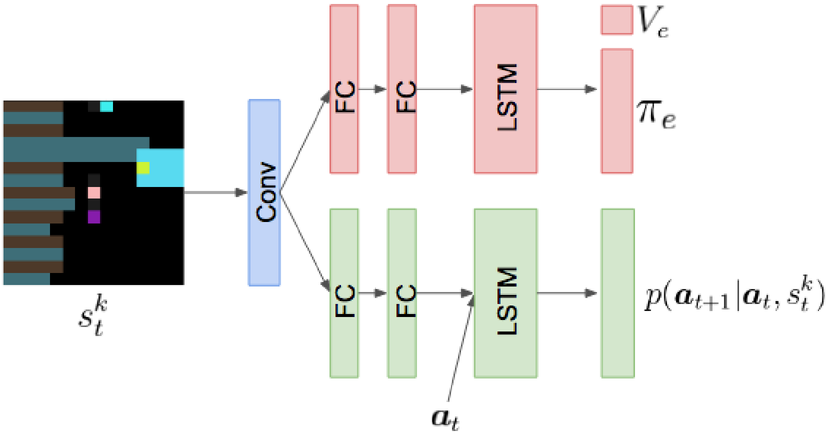


Figure 3.8: The Model of Other Agents (MOA) architecture learns both an RL policy π_e, and a supervised model that predicts the actions of other agents, $\boldsymbol{a}_{t+1}$. The supervised model is used for internally computing the influence reward.

A trained MOA can be used to compute the social influence reward using Algorithm 3. Each agent can "imagine" counterfactual actions that it could have taken at each timestep, and use its internal MOA to predict the effect on other agents. It can then give itself reward for taking actions that it estimates were the most influential. This has an intuitive appeal, because it resembles how humans reason about their effect on others (Ferguson et al., 2010). We often find ourselves asking counterfactual questions of the form, "How would she have acted if I had done something else in that situation?", which we answer using our internal model of others.

Algorithm 3: Computing the influence reward for agent k using a trained model of other agents (MOA).

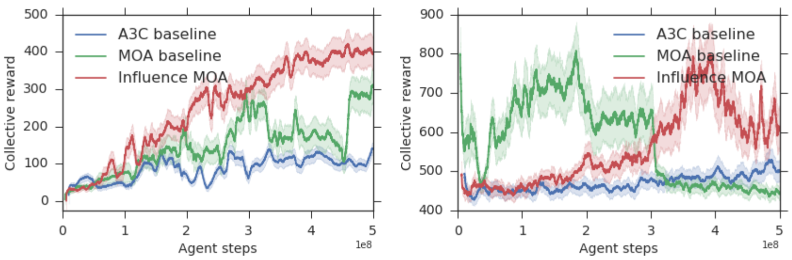
```
Require: $\tau$, trajectory containing actions for all agents, but only the observations and rewards of agent $k$. Let $T$ be the trajectory length.
Require: A trained MOA parameterized by $\theta^k_M$
influence = $[0] * T$
for timestep $t$ in $[0, T)$ do
    for other agent $j$ in $N$ do
        prob_aj = 0
        for action $\tilde{a}^k_t$ in $\mathcal{A}$ do
            Compute $p(\tilde{a}^k_{t+1} | \boldsymbol{a}_t, s^k_t; \theta^k_\pi)$ using agent $k$'s policy network
            Compute $p(a^j_{t+1} | \tilde{a}^k_t, s^k_t; \theta^k_M)$ using MOA
            prob_aj = prob_aj + $p(a^j_{t+1} | \tilde{a}^k_t, s^k_t) p(\tilde{a}^k_t | \boldsymbol{a}_t, s^k_t)$
        end
        $p(a^j_{t+1} | s^j_t) \leftarrow$ prob_aj
        influence[$t$] = influence[$t$] + $D_{KL}[p(a^j_t | a^k_t, s^k_t; \theta^k_M) || p(a^j_t | s^j_t)]$
    end
end
```

Learning a model of $p(a^j_{t+1} | a^k_t, s^k_t)$ requires implicitly modeling both other agents' internal states and behavior, as well as the environment transition function. If the model is inaccurate, this would lead to noisy estimates of the causal influence reward. To compensate for this, We only give the influence reward to an agent (k) when the agent it is attempting to influence (j) is within its field-of-view, because the estimates of $p(a^j_{t+1} | a^k_t, s^k_t)$ are more accurate when j is visible to k.[2] This constraint could have the side-effect of encouraging agents to stay in closer proximity. However, an intrinsic social reward encouraging proximity is reasonable given that humans seek affiliation and to spend time near other people (Tomasello, 2009b).

[2] This contrasts with our previous models in which the influence reward was obtained even from non-visible agents.

3.6.1 Experiment III: Modeling other agents

As before, we allow the policy LSTM of each agent to condition on the actions of other agents in the last timestep (actions are visible). We compare against an ablated version of the architecture shown in Figure 3.8, which does not use the output of the MOA to compute a reward; rather, the MOA can be thought of as an unsupervised auxiliary task that may help the model to learn a better shared embedding layer, encouraging it to encode information relevant to predicting other agents' behavior. Figure 3.9 shows the collective reward obtained for agents trained with a MOA module. While we see that the auxiliary task does help to improve reward over the A3C baseline, the influence agent gets consistently higher collective reward. These results demonstrate that the influence reward can be effectively computed using an internal MOA, and thus agents can learn socially but independently, optimizing for a social reward without a centralized controller.



(a) *Cleanup*

(b) *Harvest*

Figure 3.9: Total collective reward for MOA models. Again, intrinsic influence consistently improves learning, with the powerful A3C agent baselines not being able to learn.

Agents with influence achieve higher collective reward than the previous state-of-the-art for these environments (275 for *Cleanup* and 750 for *Harvest*) (Hughes et al., 2018). This is compelling, given that previous work relied on the assumption that agents could view one another's rewards; we make no such assumption, instead relying only on agents viewing each other's actions. Table 3.4 of the Supplementary Material gives the final collective reward obtained in previous work, and by each influence model for all three experiments.

3.7 Related work

Several attempts have been made to develop intrinsic social rewards.[3]

---

[3] Note that *intrinsic* is not a synonym of *internal*; other people can be intrinsically motivating (Stavropoulos and Carver, 2013).

Sequeira et al. (2011) developed hand-crafted rewards for a foraging environment, in which agents were punished for eating more than their fair share of food. Another approach gave agents an emotional intrinsic reward based on their perception of their neighbours' cooperativeness in a networked version of the iterated prisoner's dilemma, but is limited to scenarios in which it is possible to directly classify each action as cooperative or non-cooperative (Yu et al., 2013). This is untenable in complex settings with long-term strategies, such as the SSDs under investigation here.

Some approaches allow agents to view each others' rewards in order to optimize for collective reward (e.g. Kleiman-Weiner et al. (2016)). Peysakhovich and Lerer (2018) show that if even a single agent is trained to optimize for others' rewards, it can significantly help the group. Hughes et al. (2018) introduced an inequity aversion motivation, which penalized agents if their rewards differed too much from those of the group. Liu et al. (2014) train agents to learn their own optimal reward function in a cooperative, multi-agent setting with known group reward. However, the assumption that agents can view and optimize for each others' rewards may be unrealistic. Thus, recent work explores training agents that learn when to cooperate based solely on their own past rewards (Peysakhovich and Lerer, 2017).

Training agents to learn emergent communication protocols has been explored (Foerster et al., 2016; Cao et al., 2018; Choi et al., 2018; Lazaridou et al., 2018; Bogin et al., 2018), with many authors finding that selfish agents do not learn to use an ungrounded, *cheap talk* communication channel effectively. Crawford and Sobel (1982) find that in theory, the information communicated is proportional to the amount of common interest; thus, as agents' interests diverge, no communication is to be expected. And while communication can emerge when agents are prosocial (Foerster et al., 2016; Lazaridou et al., 2018), curious (Oudeyer and Kaplan, 2006; Oudeyer and Smith, 2016; Forestier and Oudeyer, 2017), or hand-crafted (Crandall et al., 2017), self-interested agents do not to learn to communicate (Cao et al., 2018). We have shown that the social influence reward can encourage agents to learn to communicate more effectively in complex environments.

Our MOA is related to work on machine theory of mind (Rabinowitz et al., 2018; Shum et al., 2019). Rabinowitz et al. (2018) demonstrated that a model trained to predict agents' actions can model false beliefs. LOLA agents model the impact of their policy on the parameter updates of other agents, and directly incorporate this into the agent's own learning rule (Foerster et al., 2018).

Barton et al. (2018) propose causal influence as a way to measure coordination between agents, specifically using Convergence Cross Mapping (CCM) to analyze the degree of dependence between two agents' policies. The limitation if CCM is that estimates of causality are known to degrade in the presence of stochastic effects (Tajima et al., 2015). Counterfactual reasoning has also been used in a multi-agent setting, to marginalize out the effect of one agent on a predicted global value function estimating collective reward, and thus obtain an improved baseline for computing each agent's advantage function (Foerster et al., 2017). A similar paper shows that counterfactuals can be used with potential-based reward shaping to improve credit assignment for training a joint policy in multi-agent RL (Devlin et al., 2014). However, once again these approaches rely on a centralized controller.

Mutual information (MI) has been explored as a tool for designing social rewards. Strouse et al. (2018) train agents to optimize the MI between their actions and a categorical goal, as a way to signal or hide the agent's intentions. However, this approach depends on agents pursuing a known, categorical goal. Guckelsberger et al. (2018), in pursuit of the ultimate video game adversary, develop an agent that maximizes its empowerment, minimizes the player's empowerment, and maximizes its empowerment over the player's next state. This third goal, termed *transfer empowerment*, is obtained by maximizing the MI between the agent's actions and the player's future state. While a social form of empowerment, Guckelsberger et al. (2018) find that agents trained with transfer empowerment simply tend to stay near the player. Further, the agents are not trained with RL, but rather analytically compute these measures in simple grid-world environments. As such, the agent cannot learn to model other agents or the environment.

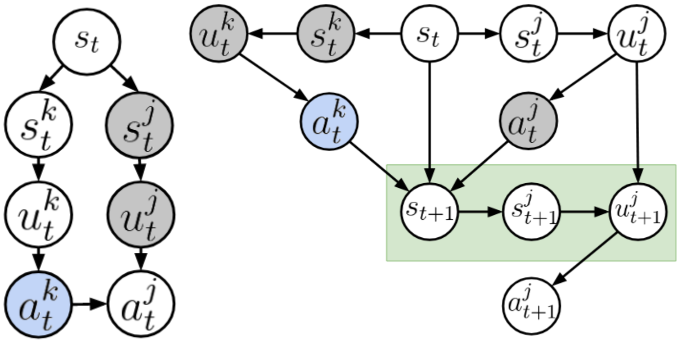
Given the social influence reward incentivizes maximizing the mutual information between agents' actions, our work also has ties to the literature on empowerment, in which agents maximize the mutual information between their actions and their future state (Klyubin et al., 2005; Mohamed and Rezende, 2015). Thus, our proposed reward can be seen as a novel social form of empowerment.

3.8 *Details on causal inference*

The causal influence reward presented in Eq. 3.1 is assessed using counterfactual reasoning. Unlike a *do*-calculus intervention (which estimates the general expected causal effect of one variable on another), a counterfactual involves conditioning on a set of variables observed in a given situation and asking how would the outcome have changed if some variable were different, and all other variables remained the same (Pearl et al., 2016). This type of inquiry allows us to measure the precise causal effect of agent k's action at timestep t, a_t^k, on agent j's action, a_t^j, in the specific environment state s_t, providing a richer and less sparse reward for agent k. Computing counterfactuals requires conditioning on the correct set of observed variables to ensure there are no confounds. In our case, the conditioning set must include not only an agent's partially observed view of the environment state, s_t^j, but also the agent's internal LSTM state u_t^j, to remove any dependency on previous timesteps in the trajectory. Thus, the basic causal influence reward can be more accurately written:

$$c_t^k = \sum_{j=0, j\neq k}^{N} \Big[D_{KL}[p(a_t^j \mid a_t^k, s_t^j, u_t^j) || p(a_t^j \mid s_t^j, u_t^j)] \Big]. \tag{3.3}$$

Figure 3.10 shows the causal diagrams for computing the influence reward in both the basic case and the MOA case. Because basic influence looks at influence between agents' actions in the same timestep, the diagram is much simpler. However, to avoid circular dependencies in the graph, it requires that agent k choose its action before j, and therefore k can influence j but j cannot influence k. If there are more than two agents, we assume a disjoint set of influencer and influencee agents, and all influencers must act first.



(a) Basic (b) MOA

Figure 3.10: Causal diagrams of agent k's effect on j's action. Shaded nodes are conditioned on, and we intervene on a_t^k (blue node) by replacing it with counterfactuals. Nodes with a green background must be modeled using the MOA module. Note that there is no backdoor path between a_t^k and s_t in the MOA case, since it would require traversing a collider that is not in the conditioning set.

Computing influence across timesteps, as in the communication and

MOA experiments, complicates the causal diagram, but ensures that each agent can influence every other agent. Figure 3.10 (b) shows the diagram in the MOA case, in which we can isolate the causal effect of a_t^k on a_{t+1}^j because the back-door path through s_t is blocked by the collider nodes at s_{t+1} and u_{t+1}^j (Pearl et al., 2016). Note that it would be sufficient to condition only on s_t^k in order to block all back-door paths in this case, but we show $\langle u_t^k, s_t^k, a_t^j \rangle$ as shaded because all of these are given as inputs to the MOA to help it predict a_{t+1}^j. For the MOA to accurately estimate $p(a_{t+1}^j | a_t^k, s_t^k)$, it must model both the environment transition function T, as well as aspects of the internal LSTM state of the other agent, u_{t+1}^j, as shown by the shaded green variables in Figure 3.10 (b).

This is a simple case of counterfactual reasoning, that does not require using abduction to update the probability of any unobserved variables (Pearl, 2013). This is because we have built all relevant models, know all of their inputs, and can easily store the values for those variables at every step of the trajectory in order to condition on them so that there are no unobserved variables that could act as a confounder.

3.9 *Conclusions and future work*

All three experiments have shown that the proposed intrinsic social influence reward consistently leads to higher collective return. Despite variation in the tasks, hyper-parameters, neural network architectures and experimental setups, the learning curves for agents trained with the influence reward are significantly better than the curves of powerful agents such as A3C and their improved baselines. In some cases, it is clear that without influence, agents fail to demonstrate any evidence of coordination, attesting to the promise of this idea and highlighting the complexity of learning general deep neural network multi-agent policies.

Experiment I also showed that the influence reward can lead to the emergence of communication protocols. In experiment II, which included an explicit communication channel, we saw that influence improved communication. Experiment III showed that influence can be computed by augmenting agents with an internal model of other agents. The influence reward can thus be computed without having access to another agent's reward function, or requiring a centralized controller. We were able to surpass state-of-the-art performance on the SSDs studied here, despite the fact that previous work relied on agents' ability to view other agents' rewards.

Using counterfactuals to allow agents to understand the effects of their actions on others is a promising approach with many extensions. Agents could use counterfactuals to develop a form of 'empathy', by simulating how their actions affect another agent's value function. Influence could also be used to drive coordinated behavior in robots attempting to do cooperative manipulation and control tasks. Finally, if we view multi-agent networks as single agents, influence could be used as a regularizer to encourage different modules of the network to integrate information from other networks; for example, to hopefully prevent collapse in hierarchical RL.

3.10 Statement of contributions

I originally conceived the idea for this work was as an answer to a general exam question set by Nando de Freitas about how to develop new forms of intrinsic motivation using inspiration from human social and emotional motivation. I developed the formulation of influence described in Equations 4.1 and 4.2, wrote the agent code, and devised and ran the experiments. Angeliki Lazaridou conducted the analysis of the learned communication protocols presented in Section 3.5.1. Ed Hughes and Joel Leibo developed the SSD environments and code to run vanilla A3C agents in those environments. Ed, Joel, Nando, and Caglar Gulcehre advised on the project throughout, along with DJ Strouse and Pedro Ortega. DJ and Pedro helped develop the proof of the connection with Mutual Information, and Pedro advised on Pearl's notion of causality and checked the correctness of Figure 3.10.

3.11 Appendix

3.11.1 Influence as Mutual Information

The causal influence of agent k on agent j is:

$$D_{KL}\left[p(a_t^j \mid a_t^k, z_t)\middle\|p(a_t^j \mid z_t)\right], \tag{3.4}$$

where z_t represents all relevant u and s background variables at timestep t. The influence reward to the mutual information (MI) between the actions of agents k and j, which is given by

$$\begin{aligned} I(A^j;A^k|z) &= \sum_{a^k,a^j} p(a^j,a^k|z)\log\frac{p(a^j,a^k|z)}{p(a^j|z)p(a^k|z)} \\ &= \sum_{a^k} p(a^k|z) D_{\mathrm{KL}}\Big[p(a^j|a^k,z)\Big\|p(a^j|z)\Big], \end{aligned} \tag{3.5}$$

where we see that the D_{KL} factor in Eq. 3.5 is the causal influence reward given in Eq. 3.4.

By sampling N independent trajectories τ_n from the environment, where k's actions a_n^k are drawn according to $p(a^k|z)$, we perform a Monte-Carlo approximation of the MI (see e.g. Strouse et al. (2018)),

$$\begin{aligned} I(A^k;A^j|z) &= \mathbb{E}_\tau\Big[D_{\mathrm{KL}}\big[p(A^j|A^k,z)\big\|p(A^j|z)\big]\Big|z\Big] \\ &\approx \frac{1}{N}\sum_n D_{\mathrm{KL}}\big[p(A^j|a_n^k,z)\big\|p(A^j|z)\big]\,. \end{aligned} \tag{3.6}$$

Thus, in expectation, the social influence reward is the MI between agents' actions.

Whether the policy trained with Eq. 3.4 actually learns to approximate the MI depends on the learning dynamics. We calculate the intrinsic social influence reward using Eq. 3.4, because unlike Eq. 3.5, which gives an estimate of the symmetric bandwidth between k and j, Eq. 3.4 gives the directed causal effect of the specific action taken by agent k, a_t^k. We believe this will result in an easier reward to learn, since it allows for better credit assignment; agent k can more easily learn which of its actions lead to high influence.

The connection to mutual information is interesting, because a frequently used intrinsic motivation for single agent RL is *empowerment*, which rewards the agent for having high mutual information between its actions and the future state of the environment (e.g. Klyubin et al. (2005); Capdepuy et al. (2007)). To the extent that the social influence reward approximates the MI, k is rewarded for having empowerment over j's actions.

The social influence reward can also be computed using other divergence measures besides KL-divergence. Lizier and Prokopenko (2010) propose *local information flow* as a measure of direct causal effect; this is equivalent to the *pointwise mutual information* (the innermost term of Eq. 3.6), given by:

$$pmi(a^k; a^j \mid Z = z) = \log \frac{p(a^j \mid a^k, z)}{p(a^j \mid z)}$$

$$= \log \frac{p(a^k, a^j \mid z)}{p(a^k \mid z)p(a^j \mid z)}. \qquad (3.7)$$

The PMI gives us a measure of influence of a single action of k on the single action taken by j. The expectation of the PMI over $p(a^j, a^k|z)$ is the MI. We experiment with using the PMI and a number of divergence measures, including the Jensen-Shannon Divergence (JSD), and find that the influence reward is robust to the choice of measure.

3.11.2 *Additional experiment - Box Trapped*

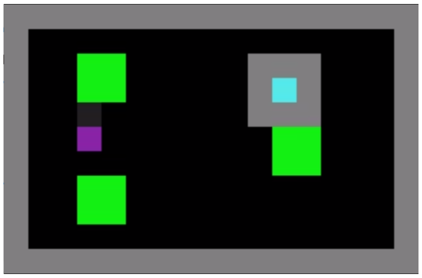


Figure 3.11: The *Box trapped* environment in which the teal agent is trapped, and the purple agent can release it with a special *open box* action.

As a proof-of-concept experiment to test whether the influence reward works as expected, we constructed a special environment, shown in Figure 3.11. In this environment, one agent (teal) is trapped in a box. The other agent (purple) has a special action it can use to open the box... or it can simply choose to consume apples, which exist outside the box and are inexhaustible in this environment.

As expected, a vanilla A3C agent learns to act selfishly; the purple agent will simply consume apples, and chooses the *open box* action in 0% of trajectories once the policy has converged. A video of A3C agents trained in this environment is available at: `https://youtu.be/C8SE9_YKzxI`, which shows that the purple agent leaves its compatriot trapped in the box throughout the trajectory.

In contrast, an agent trained with the social influence reward chooses the *open box* action in 88% of trajectories, releasing its fellow agent so that they are both able to consume apples. A video of this behavior is shown at: `https://youtu.be/Gfo248-qt3c`. Further, as Figure 3.12 reveals, the purple influencer agent usually chooses to open the box within the first few steps of the trajetory, giving its fellow agent more time to collect reward.

Most importantly though, Figure 3.13 shows the influence reward over the course of a trajectory in the *Box trapped* environment. The agent chooses the *open box* action in the second timestep; at this point, we see a corresponding spike in the influence reward. This reveals that the influence reward works as expected, incentivizing an action which has a strong—and in this case, prosocial—effect on the other agent's behavior.

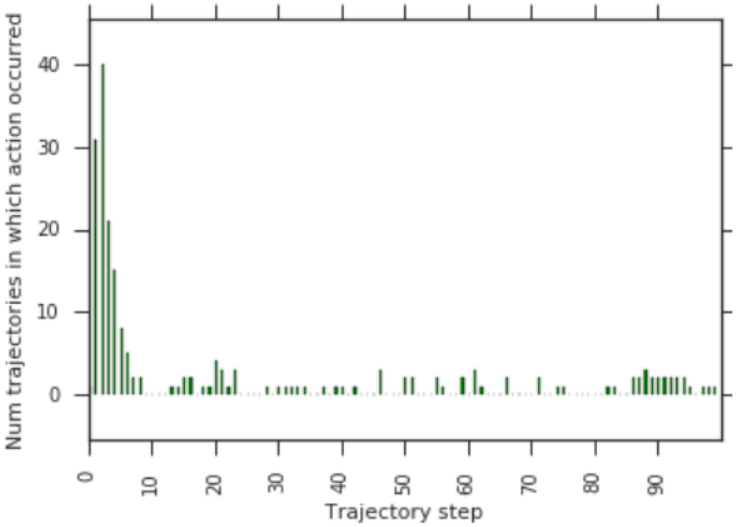


Figure 3.12: Number of times the *open box* action occurs at each trajectory step over 100 trajectories.

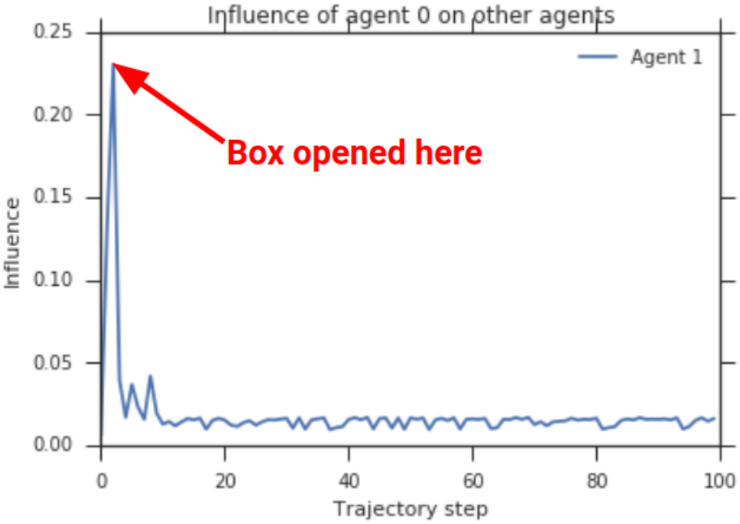


Figure 3.13: Influence reward over a trajectory in *Box trapped*. An agent gets high influence for letting another agent out of the box in which it is trapped.

3.11.3 *Implementation details*

All models are trained with a single convolutional layer with a kernel of size 3, stride of size 1, and 6 output channels. This is connected to two fully connected layers of size 32 each, and an LSTM with 128 cells. We use a discount factor $\gamma = .99$. The number of agents N is fixed to 5.

In addition to the comparison function used to compute influence (e.g. KL-divergence, PMI, JSD), there are many other hyperparameters that can be tuned for each model. We use a random search over hyperparameters, ensuring a fair comparison with the search size over the baseline parameters that are shared with the influence models. For all models we search for the optimal entropy reward and learning rate, where we anneal the learning rate from an initial value `lr_init` to `lr_final`. The below sections give the parameters found to be most effective for each of the three experiments.

3.11.3.1 *Basic influence hyperparameters*

In this setting we vary the number of influencers from $1-4$, the influence reward weight β, and the number of curriculum steps over which the weight of the influence reward is linearly increased C. In this setting, since we have a centralised controller, we also experiment with giving the influence reward to the agent being influenced as well, and find that this sometimes helps. This 'influencee' reward is not used in the other two experiments, since it precludes independent training. The hyperparameters found to give the best performance for each model are shown in Table 3.1.

	Cleanup			Harvest		
Hyperparameter	A3C baseline	Visible actions baseline	Influence	A3C baseline	Visible actions baseline	Influence
Entropy reg.	.00176	.00176	.000248	.000687	.00184	.00025
lr_init	.00126	.00126	.00107	.00136	.00215	.00107
lr_end	.000012	.000012	.000042	.000028	.000013	.000042
Num. influencers	-	3	1	-	3	3
Influence weight β	-	0	.146	-	0	.224
Curriculum C	-	-	140	-	-	140
Policy comparison	-	-	JSD	-	-	PMI
Influencee reward	-	-	1	-	-	0

Table 3.1: Optimal hyperparameter settings for the models in the basic influence experiment.

3.11.3.2 *Communication hyperparameters*

Because the communication models have an extra A2C output head for the communication policy, we use an additional entropy regularization term just for this head, and apply a weight to the communication loss in the loss function. We also vary the number of communication symbols that the agents can emit, and the size of the linear layer that connects the LSTM to the communication policy layer, which we term the communication embedding size. Finally, in the communication regime, we experiment to setting the weight on the extrinsic reward E, α, to zero. The best hyperparameters for each of the communication models are shown in Table 3.2.

Hyperparameter	Cleanup			Harvest		
	A3C baseline	Comm. baseline	Influence comm.	A3C baseline	Comm. baseline	Influence comm.
Entropy reg.	.00176	.000249	.00305	.000687	.000174	.00220
lr_init	.00126	.00223	.00249	.00136	.00137	.000413
lr_end	.000012	.000022	.0000127	.000028	.0000127	.000049
Influence weight β	-	0	2.752	-	0	4.825
Extrinsic reward weight α	-	-	0	-	-	1.0
Curriculum C	-	-	1	-	-	8
Policy comparison	-	-	KL	-	-	KL
Comm. entropy reg.	-	-	.000789	-	-	.00208
Comm. loss weight	-	-	.0758	-	-	.0709
Symbol vocab size	-	-	9	-	-	7
Comm. embedding	-	-	32	-	-	16

Table 3.2: Optimal hyperparameter settings for the models in the communication experiment.

3.11.3.3 *Model of other agents (MOA) hyperparameters*

The MOA hyperparameters include whether to only train the MOA with cross-entropy loss on the actions of agents that are visible, and how much to weight the supervised loss in the overall loss of the model. The best hyperparameters are shown in Table 3.3.

Hyperparameter	Cleanup			Harvest		
	A3C baseline	MOA baseline	Influence MOA	A3C baseline	MOA baseline	Influence MOA
Entropy reg.	.00176	.00176	.00176	.000687	.00495	.00223
lr_init	.00126	.00123	.00123	.00136	.00206	.00120
lr_end	.000012	.000012	.000012	.000028	.000022	.000044
Influence weight β	-	0	.620	-	0	2.521
MOA loss weight	-	1.312	15.007	-	1.711	10.911
Curriculum C	-	-	40	-	-	226
Policy comparison	-	-	KL	-	-	KL
Train MOA only when visible	-	False	True	-	False	True

Table 3.3: Optimal hyperparameter settings for the models in the model of other agents (MOA) experiment.

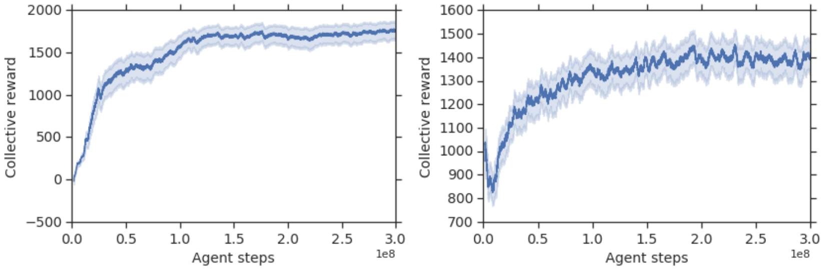
3.11.3.4 *Communication analysis*

The speaker consistency metric is calculated as:

$$\sum_{k=1}^{N} 0.5[\sum_{c} 1 - \frac{H(p(a^k|m^k = c))}{H_{max}} + \sum_{a} 1 - \frac{H(p(m^k|a^k = a))}{H_{max}}], \tag{3.8}$$

where H is the entropy function and H_{max} is the maximum entropy based on the number of discrete symbols or actions. The goal of the metric is to measure how much of a 1:1 correspondence exists between a speaker's action and the speaker's communication message.

3.11.4 *Additional results*



(a) *Cleanup* (b) *Harvest*

Figure 3.14: Total collective reward obtained by agents trained to optimize for the collective reward, for the 5 best hyperparameter settings with 5 random seeds each. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps.

In this section we include the results of training explicitly prosocial agents, which directly optimize for the collective reward of all agents. Previous work (e.g. Peysakhovich and Lerer (2018)) has shown that training agents to optimize for the rewards of other agents can help the group to obtain better collective outcomes. Following a similar principle, we implemented agents that optimize for a convex combination of their own individual reward e_t^k and the collective reward of all other agents, $\sum_{i=1,i\neq k}^{N} e_t^i$. Thus, the reward function for agent k is $r_t^k = e_t^k + \eta \sum_{i=1,i\neq k}^{N} e_t^i$. We conducted the same hyperparameter search over the parameters mentioned in Section 3.11.3.1 varying the weight placed on the collective reward, $\eta \in [0, 2]$.

As expected, we find that agents trained to optimize for collective reward attain higher collective reward in both *Cleanup* and *Harvest*, as is shown in Figure 3.14. In both games, the optimal value for $\eta = 0.85$. Interestingly, however, the equality in the individual returns for these agents is extremely low. Across the hyperparameter sweep, no solution to the *Cleanup* game which scored more than 20 points in terms of collective return was found in which all agents scored an individual return above 0. It seems that in *Cleanup*, when agents are trained to optimize for collective return, they converge on a solution in which some agents never receive any reward.

Note that training agents to optimize for collective reward requires that each agent can view the rewards obtained by other agents. As discussed previously, the social influence reward is a novel way to obtain cooperative behavior, that does not require making this assumption.

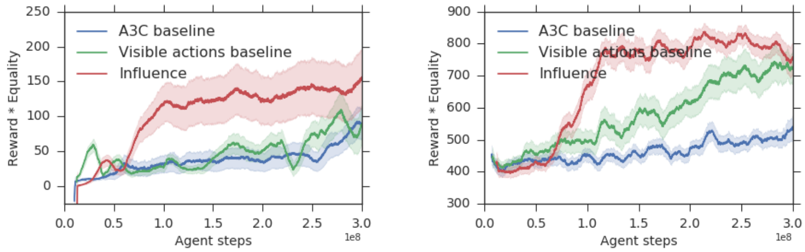
3.11.4.1 *Collective reward and equality*

It is important to note that collective reward is not always the perfect metric of cooperative behavior, a finding that was also discovered by Barton et al. (2018) and emphasized by Leibo et al. (2017). In the case, we find that there is a spurious solution to the *Harvest* game, in which one agent fails to learn and fails to collect any apples. This leads to very high collective reward, since it means there is one fewer agent that can exploit the others, and makes sustainable harvesting easier to achieve. Therefore, for the results shown in here, we eliminate any random seed in *Harvest* for which one of the agents has failed to learn to collect apples, as in previous work (Hughes et al., 2018).

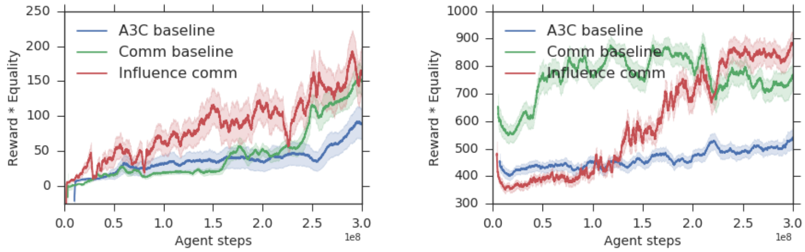
However, here we also present an alternative strategy for assessing the overall collective outcomes: weighting the total collective reward by an index of equality of the individual rewards. Specifically, we compute the Gini coefficient over the N agents' individual environmental rewards e_t^k:

$$G = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |e_t^i - e_t^j|}{2N \sum_{i=1}^{N} e_t^i}, \tag{3.9}$$

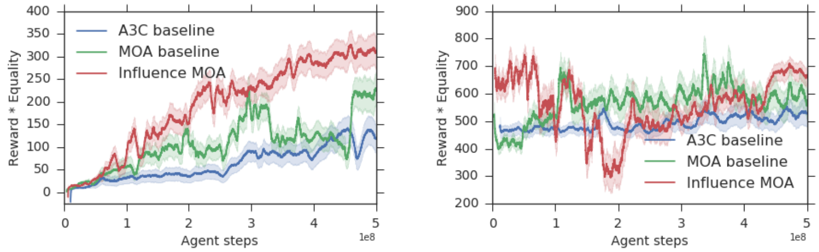
which gives us a measure of the inequality of the returns, where $G \in [0, 1]$, with $G = 0$ indicating perfect equality. Thus, $1 - G$ is a measure of equality; we use this to weight the collective reward for each experiment, and plot the results in Figure 3.15. Once again, we see that the influence models give the highest final performance, even with this new metric.



(a) *Cleanup* - Basic influence (b) *Harvest* - Basic influence



(c) *Cleanup* - Communication (d) *Harvest* - Communication



(e) *Cleanup* - Model of other agents (f) *Harvest* - Model of other agents

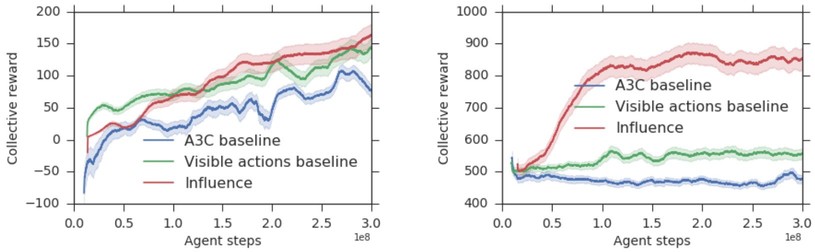
Figure 3.15: Total collective reward times equality, $R * (1 - G)$, obtained in all experiments. Error bars show a 99.5% confidence interval (CI) over 5 random seeds, computed within a sliding window of 200 agent steps. Once again, the models trained with influence reward (red) significantly outperform the baseline and ablated models.

3.11.4.2 *Collective reward over multiple hyperparameters*

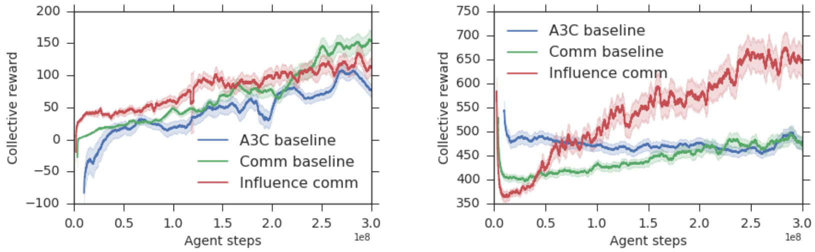
Finally, we would like to show that the influence reward is robust to the choice of hyperparameter settings. Therefore, in Figure 3.16, we plot the collective reward of the top 5 best hyperparameter settings for each experiment, over 5 random seeds each. Once again, the influence models result in higher collective reward, which provides evidence that the model is robust to the choice of hyperparameters.

3.11.4.3 *Performance comparison between models and related work*

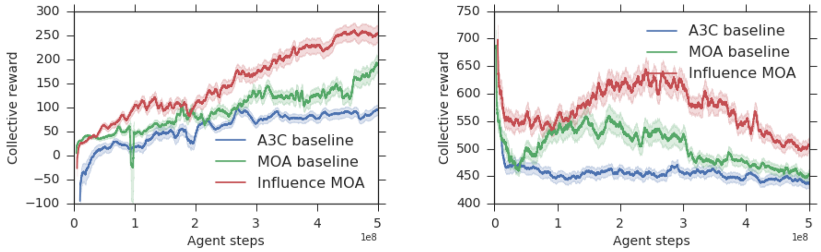
Table 3.4 presents the final collective reward obtained by each of the models tested in the experiments presented. We see that in several cases, the influence agents are even able to out-perform the state-of-



(a) *Cleanup* - Basic influence (b) *Harvest* - Basic influence



(c) *Cleanup* - Communication (d) *Harvest* - Communication



(e) *Cleanup* - Model of other agents (f) *Harvest* - Model of other agents

Figure 3.16: Total collective reward over the top 5 hyperparameter settings, with 5 random seeds each, for all experiments. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps. The influence models still maintain an advantage over the baselines and ablated models, suggesting the technique is robust to the hyperparameter settings.

the-art results on these tasks reported by Hughes et al. (2018). This is impressive, considering the *inequity averse* agents are able to view all other agents' rewards. We make no such assumption (only assuming that agents can view each others' actions), and yet are able to achieve similar or superior performance.

	Cleanup	Harvest
A3C baseline	89	485
Inequity aversion (Hughes et al.)	275	750
Influence - Basic	190	**1073**
Influence - Communication	166	**951**
Influence - Model of other agents	**392**	588

Table 3.4: Final collective reward over the last 50 agent steps for each of the models considered. Bolded entries represent experiments in which the influence models significantly outperformed the scores reported in previous work on *inequity aversion* (Hughes et al., 2018)

4 *Learning from implicit human preferences in dialog*

4.1 *Interactive human evaluation of dialog systems* 134
4.1.1 *Background* 135
4.1.2 *Related work* 137
4.1.3 *Knowledge distillation for sentiment and semantic regularization* 138
4.1.4 *Evaluation methodologies* 140
4.1.5 *Details about implicit metrics* 142
4.1.6 *Experiments* 144
4.1.7 *Conclusions* 148
4.2 *Way off-policy batch reinforcement learning* 149
4.2.1 *Background* 150
4.2.2 *Related Work* 152
4.2.3 *RL for language generation* 154
4.2.4 *Methods* 154
4.2.5 *Traditional RL experiments* 158
4.2.6 *Batch RL for learning dialog from human feedback* 160
4.2.7 *Dialog results* 162
4.2.8 *Conclusion* 166
4.3 *KL-control for improved sequence generation with RL* 167
4.3.1 *Recurrent G-learning* 167
4.3.2 *KL-control for sequence generation* 168
4.3.3 *Experiment I: Melody generation* 168
4.3.4 *Experiment II: Computational drug discovery* 174
4.3.5 *Conclusion and future work* 175
4.4 *Statement of contributions* 176
4.5 *Appendix* 177
4.5.1 *Self-play hybrid metric coefficients* 177
4.5.2 *Reddit casual conversation corpus details* 178
4.5.3 *Embedding-based metrics* 179
4.5.4 *RL post-hoc metrics* 179

4.5.5 Interactive evaluation details 180
4.5.6 Website server setup and configuration 182
4.5.7 Hyper-parameter tuning details 183
4.5.8 One-turn evaluation experiment details 185
4.5.9 Additional results 186

In order to create AI systems that not only provide benefit to humans, but that are aligned with long-term human interests, it is important to develop techniques for automatically learning from human preferences. This chapter explores how to create a neural network dialog model that can learn by conversing with a human. Specifically, we focus on learning from implicit cues in the text itself (such as the sentiment expressed), in order to improve the model based on human feedback.

This chapter has three parts. Section 4.1 addresses the problem of building and evaluating open-domain dialog models via interactive human evaluation. Building from the data collected in these first experiments, Section 4.2 describes how to learn from a fixed, static batch of this human data using reinforcement learning (RL), in the unusual setting where the model is not able to explore online in the environment. Finally, Section 4.3 shows that the same techniques can be applied to a variety of related sequence modeling problems, yielding good results, and reducing catastrophic forgetting when performing RL via transfer learning from a model pre-trained on data.

4.1 Interactive human evaluation of dialog systems

Building an open-domain conversational agent is a challenging problem. Current evaluation methods, mostly post-hoc judgments of single-turn evaluation, do not capture conversation quality in a realistic interactive context. Therefore we investigate interactive human evaluation and provide evidence for its necessity; we then introduce a novel, model-agnostic, and dataset-agnostic method to approximate it. In particular, we propose a self-play scenario where the dialog system talks to itself and we calculate a combination of proxies such as sentiment and semantic coherence on the conversation trajectory. We show that this metric is capable of capturing the human-rated quality of a dialog model better than any automated metric known to-date, achieving a significant Pearson correlation ($r > .7, p < .05$). To investigate the strengths of this novel metric and interactive evaluation in comparison to state-of-the-art metrics and one-turn evaluation, we perform extended experiments with a set of models, including several that make novel improvements to recent hierarchical dialog generation architectures through sentiment and semantic knowledge distillation on the utterance level. Finally, we open-source the interactive evaluation platform we built and the dataset we collected to allow researchers to efficiently deploy and evaluate generative dialog models.

4.1.1 Background

The goal of an open-domain conversational agent is to carry out natural social interactions with humans. Current state-of-the-art generative neural networks fail in producing key aspects of good natural conversation, including staying on topic, not being repetitive, and generating emotionally appropriate responses. One of the biggest challenges in training better dialog systems relates to the difficulty of evaluating them. Automatic metrics such as BLEU score relate poorly to human judgment of dialog quality (Liu et al., 2016), and while embedding-distance based metrics provide an alternative (Mitchell and Lapata, 2008), we will show that they also do not correlate well with human evaluation. Without a reliable metric to optimize, training high quality dialog models remains difficult.

Since humans are the ultimate authority on what constitutes a good conversation, many authors rely on human ratings to evaluate their methods (e.g. (Serban et al., 2017b; Park et al., 2018; Serban et al., 2016)). The predominant procedure for obtaining human ratings uses *single-turn evaluation*: a context of several sentences is used as input to generate a single response for a human to evaluate. However, a single turn is incapable of capturing common failure modes of open-domain dialog systems, such as a lack of diversity in the responses, inability to track long-term aspects of the conversation, and inability to maintain a consistent persona (Dinan et al., 2019; Li et al., 2016a; Mazare et al., 2018). In spite of this, most authors still rely on single-turn evaluation, even when evaluating these exact qualities.

Conversation is inherently a process. In this chapter, we argue that multi-turn interactive human evaluation is essential for evaluating this process, and for making progress in improving open-domain

Single-turn **HRED - Cornell**	**Multi-turn** **HRED - Cornell**	**Multi-turn** **EI HRED - Reddit**	**Self-play** **EI HRED - Reddit**
[User]: you're weird. [HRED]: i'm sorry.	[User]: i'm feeling sad today. [HRED]: no. [User]: yes i am! [HRED]: i'm sorry, i'm sorry. i don't want to know. [User]: you're weird. [HRED]: i'm sorry.	[User] talk to me [EI HRED]: it's my job. [User]: i love you [EI HRED]: thank you for the kind words. [User]: do you love me? [EI HRED]: it's a bit personal, so I sent you a dm. [User]: i'm blushing	[EI HRED] i can't wait to see what i can do. [EI HRED] that's a good idea. [EI HRED] how do you deal with people? [EI HRED] i've been on reddit for a long time. [EI HRED] i am in the same boat.

Table 4.1: Single-turn evaluation fails to capture a lack of diversity in a dialog model's responses, as well as its inability to track the conversation and respond in emotionally appropriate ways. We argue multi-turn evaluation is needed to evaluate dialog models, and show that our Emotion+Infersent (EI) models trained on a larger and more diverse corpus, produce better interactive dialog. We present strong evidence that our novel dialog self-play framework combined with psychologically motivated novel automated metrics can accurately estimate quality of a model with respect to its ability to carry out multi-turn conversations.

dialog models. Table 4.1 illustrates an example where a bot generates a coherent single-turn response, but multi-turn evaluation shows that the distribution of its responses has collapsed onto repeatedly saying the same phrase, a common problem in this domain (Li et al., 2016c).

The relative sparsity of interactive human evaluation of dialog systems may relate to the difficulty and expense of collecting human data. Therefore, we develop a way to approximate human judgment of interactive dialog quality using a novel form of dialog self-play. We begin by proposing a series of metrics to evaluate the quality of conversation motivated by findings in psychology. We then fit a function that predicts human assessments of conversation quality given these metrics. This function is used to predict bot quality through self-play: for a fixed number of turns, the bot generates utterances which are fed back into itself as input in the next turn. The same metrics described above are computed on the self-play generated conversation, and the same function fit to human data is used to predict the bot quality. We show a very high correlation ($r = .725, p = .008$) between the predicted quality scores and the ground-truth human judgments of bot quality, suggesting self-play is a good proxy for interactive conversation assessment.

To demonstrate the relevance of the interactive evaluation and the proposed self-play evaluation, we perform extended experiments with different hierarchical architectures. In particular, we compare three recent baseline hierarchical architectures: HRED, VHRED, VHCR. Motivated by sentiment and semantics being key aspects of producing high quality conversations, we regularize the top level of the hierarchy to ensure it encodes such information, using a form of model distillation (Hinton et al., 2015). Our results show the effectiveness of the proposed regularization in interactive evaluation in both the human-bot and the self-play scenarios.

This section makes three main contributions: 1) demonstrates the necessity of interactive multi-turn evaluation to capture the quality of the dialog systems; 2) Presents a novel self-play framework to estimate a new psychology-motivated hybrid quality score. These estimations are highly correlated with quality scores obtained from interactive human evaluation, more strongly than the state-of-the-art automated metrics; 3) proposes a new method of regularizing hierarchical seq2seq models with knowledge distillation. All the code, data, and interactive evaluation platform resulting from our work are publicly available.

4.1.2 *Related work*

Despite the noisiness of single-turn human evaluation, interactive evaluation in dialog has been mostly limited to presenting the results of competitions (e.g. the Alexa prize (Serban et al., 2017a; Venkatesh et al., 2018), or the Conversational Intelligence Challenge (Dinan et al., 2019)). Those findings reveal that most bots do not perform well in interactive evaluation, due to repetitiveness, inability to balance dialog acts across the conversation, and inability to maintain a consistent persona (Dinan et al., 2019). Even work aimed at maintaining a persona does not test in an interactive setting (Mazare et al., 2018; Li et al., 2016a). To the best of our knowledge, no prior work has compared interactive, multi-turn human evaluations of open-domain dialog models to traditional forms of evaluation.

Dialog systems remain difficult to train due to the lack of metrics that can effectively capture good dialog quality. Several authors have proposed training automatic predictors of human judgment or to combine human judgment with automatic metrics (Hashimoto and Sassano, 2018; Lowe et al., 2017a; Hashimoto et al., 2019). However, a state-of-the-art model trained to predict human judgments achieved a correlation of less than 0.5 with the ground truth (Lowe et al., 2017a).

Perhaps the lack of research into interactive evaluation relates to the difficulty and expense. We show that human judgments of the quality of an interactive evaluation can be automatically and reliably approximated using dialog model self-play. There is limited work investigating self-play for dialog systems: Shah et al. (2018) use a task schema and user simulator to generate samples for input to a goal-directed dialog system, while Li et al. (2016c) use a copy of a dialog model to compute a reward function that can be optimized with reinforcement learning. However, we are not aware of prior work using self-play for approximating interactive human evaluation.

Multi-turn conversation necessitates tracking long-term aspects of the dialog like the topic and tone. Hierarchical recurrent neural networks (RNNs) have been proposed as a way to improve long-term tracking of the conversation, through maintaining both a word- and utterance-level RNN (e.g. (Serban et al., 2016, 2017b; Park et al., 2018; Shen et al., 2018; Zhao et al., 2017)). Yet dialog is more than language modeling, it requires topic and social coherence. Prior performance improvements to dialog models using topic information include appending topic as an additional input (Ghosh et al., 2016), or extracting topic information using Latent Dirchlet Allocation (Li and Jurafsky, 2017; Xing et al., 2017). Towards social and emotional coherence, previous works have investigated various features and loss functions based on emotion (Zhou et al., 2018a; Zhou and Wang, 2018; Huang et al., 2018a; Rashkin et al., 2018).

4.1.3 *Knowledge distillation for sentiment and semantic regularization*

We build on three existing hierarchical seq2seq[1] architectures designed for dialog. Here, we provide a brief summary; for detailed information, see (Serban et al., 2016, 2017b; Park et al., 2018). The first baseline model, Hierarchical Recurrent Encoder Decoder (HRED) (Serban et al., 2016) extends a traditional seq2seq model by adding a third recurrent neural network (RNN), which is only updated after each dialog turn, or utterance. The idea behind this *Context RNN* is that it could potentially track longer term aspects of the conversation, such as the topic; however, there is no guarantee that it will learn to do so. The decoder of the HRED model conditions on both the embedding produced by the encoder for the current utterance, h_n^e, and the embedding of the Context RNN for the previous utterance, h_{n-1}^c.

The second baseline model, Variational HRED (VHRED) (Serban et al., 2017b), extends HRED with a variational constraint on the utterance embedding space z. Let $x_n = [w_{1n}, w_{2n} \dots w_{mn}]$ be the n-th

---

[1] Note that while transformer architectures (e.g. (Radford et al., 2019)) have emerged as a powerful alternative to seq2seq models, here we choose to focus on hierarchical architectures because it gives us the flexibility to extend our reinforcement learning methods described in Section 4.2 to use hierarchical control in the future, thus learning to optimize rewards at both the utterance and conversation level.

utterance composed of tokens $w_{1..m}$. VHRED predicts x_n as follows:

$$h_n^e = f^e(x_{n-1}) \tag{4.1}$$

$$h_{n-1}^c = f^c(x_{n-1}, h_{n-1}^e) \tag{4.2}$$

$$\mu, \Sigma = f(h_{n-1}^c) \tag{4.3}$$

$$p_\theta(z_n | x_{<n}) = N(z|\mu, \Sigma) \tag{4.4}$$

$$p(x_n | x_{<n}) = f^d(h_{n-1}^c, z_n) \tag{4.5}$$

Equations (4.1)-(4.5) describe the computation of VHRED at inference time where f^e, f^c, and f^d are Gated Recurrent Unit (GRU) networks for the encoder, context, and decoder RNNs, respectively; at training time, it allows the computation of z, μ, and Σ to condition on the encoding of the target utterance, h_n^e, giving the posterior distribution $p_\Psi(z_n | x_{\leq n})$. A Kullback-Leibler (KL) divergence constraint is placed between the posterior and prior, $D_{KL}(p_\Psi || p_\theta)$. The third model, Variational Hierarchical Conversation RNN (VHCR) (Park et al., 2018) further extends VHRED by drawing a prior encoding $z^{conv} \sim N(0, I)$ for each conversation, allowing all parts of the model (f^c, μ, Σ) to condition on z^{conv}, which is unchanging throughout the conversation.

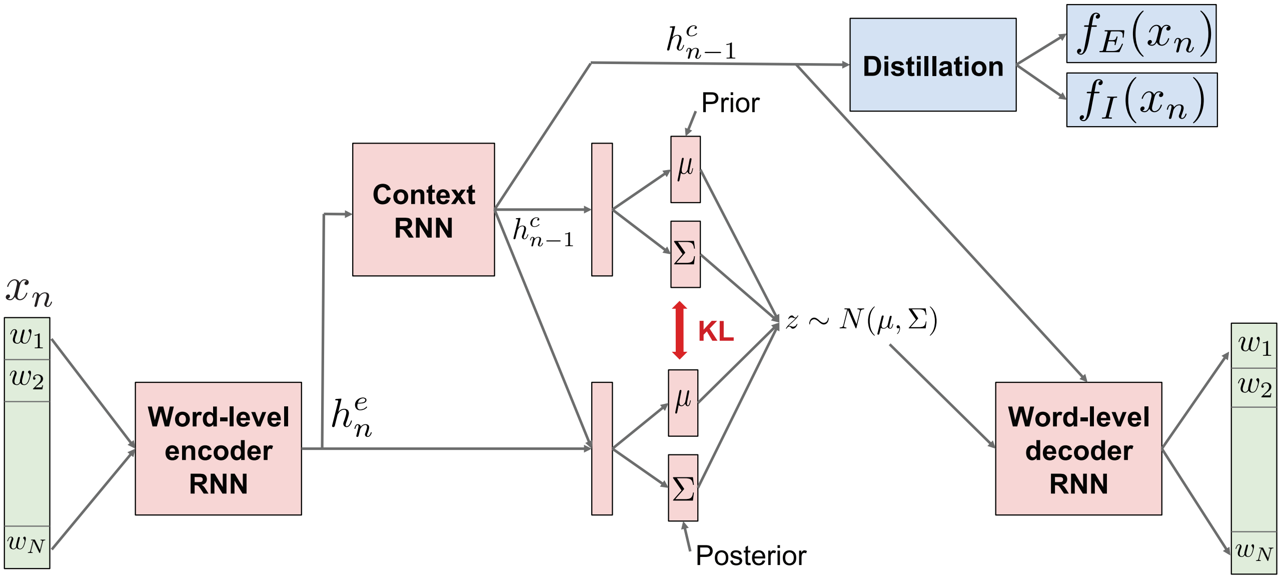


Figure 4.1: Illustration of the EI regularization (blue) applied to VHRED baseline (red) to enforce encoding sentiment and semantics of an utterance in the Context RNN. The EI regularization can be similarly applied to HRED and VHCR.

4.1.3.1 *Emotion and Infersent regularization (EI)*

While the hierarchical design of these models is motivated by a desire to allow tracking high-level, slow-changing aspects of the conversation like topic or tone, it is unclear that the network will be able to model these aspects without additional structure or information. We thus propose a regularization to the top level of the hierarchy, the Context RNN, to force it to encode both the sentiment and semantics of the utterance. To do this, we leverage a state-of-the-art sentiment detection model trained on a large Twitter corpus (Felbo et al., 2017), as well as the recently proposed *Infersent* sentence-embedding model trained to predict the meaning (i.e. entailment, contradiction) of sentences (Conneau et al., 2017), and distill them into the *Context RNN*.

First, we use these models to predict the emotional content, $f_E(x_n)$, and infersent embedding, $f_I(x_n)$ of each input utterance. We then add an additional network to the hierarchical models which predicts these values based on the context RNN embedding of the utterance: $f^{distill}(h_n^c) =< f_E(x_n), f_I(x_n) >$. The goal is to transfer knowledge of emotion and semantics in text into the context RNN via knowledge distillation (Hinton et al., 2015).

Figure 4.1 illustrates, in blue color, the EI regularization applied to the VHRED model. The regularization can be similarly applied to HRED and VHCR. In our experiments we refer to the regularized models as HRED-EI, VHRED-EI, and VHCR-EI, respectively, or, more generally, EI models as opposed to baseline models. The code for all our models is available at `https://github.com/natashamjaques/neural_chat` and was originally based on (Park et al., 2018).

4.1.4 *Evaluation methodologies*

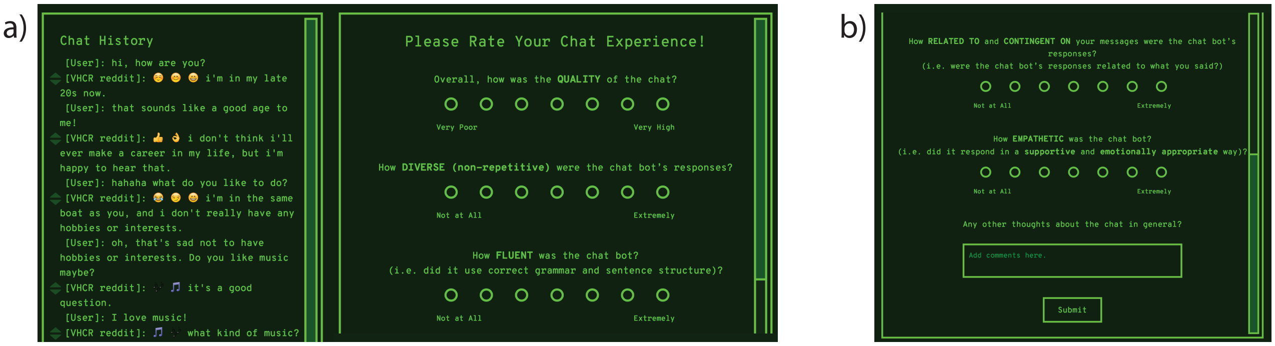
4.1.4.1 *Traditional evaluation*

Automatic metrics Embedding-based metrics compare generated sentences to ground truth sentences using a vector representation of words (Mitchell and Lapata, 2008). In this work we use three embedding metrics: embedding *average*, vector *extrema*, and *greedy* matching. These three metrics are used in previous open-domain dialog models (Serban et al., 2017b; Liu et al., 2016; Park et al., 2018). We also use *perplexity* as a standard measure of the likelihood of the generated sentences with respect to the target outputs. Another common metric for variational models is the KL-Divergence between the posterior and the prior distribution, as a way of assessing the information encoded into the latent variables (Shen et al., 2018) (Figure 4.1 illustrates KL for the VHRED model).

Conventional one-turn human evaluation We employ a similar method to previous work for our single-turn human evaluation of generated responses (Serban et al., 2017b; Park et al., 2018), sampling contexts from each corpus and asking humans to compare the generated responses. To reduce ambiguity, we exclude contexts shorter than 10 tokens and contexts containing <unknown> tokens. We recruited participants from Amazon Mechanical Turk (AMT) to compare generated sentences. Annotators could also select a third "tied" option. For each example (context and pair of generated sentences), we asked annotators to rate quality, fluency, relatedness, and empathy of the generated sentences. Each batch of 100 pairwise comparison were labeled by 6 - 8 annotators.

4.1.4.2 *Interactive human evaluation*

To address the limitations of single-turn human evaluation, we built a platform for conducting interactive evaluation of dialog models with humans, which we make available in open-source to the community (see Figure 4.2). Annotators rated quality, fluency, relatedness, and empathy of a bot after interacting with it for at least 3 turns. Participants can also upvote or downvote each bot response.



Chat History

[User]: hi, how are you?
[VHCR reddit]: ☺️ 😊 😄 i'm in my late 20s now.
[User]: that sounds like a good age to me!
[VHCR reddit]: 👍 👌 i don't think i'll ever make a career in my life, but i'm happy to hear that.
[User]: hahaha what do you like to do?
[VHCR reddit]: 😂 😏 😄 i'm in the same boat as you, and i don't really have any hobbies or interests.
[User]: oh, that's sad not to have hobbies or interests. Do you like music maybe?
[VHCR reddit]: 🎶 🎵 it's a good question.
[User]: I love music!
[VHCR reddit]: 🎵 🎶 what kind of music?

Please Rate Your Chat Experience!

Overall, how was the **QUALITY** of the chat?

○ ○ ○ ○ ○ ○ ○

Very Poor — Very High

How **DIVERSE (non-repetitive)** were the chat bot's responses?

○ ○ ○ ○ ○ ○ ○

Not at All — Extremely

How **FLUENT** was the chat bot?
(i.e. did it use correct grammar and sentence structure)?

○ ○ ○ ○ ○ ○ ○

Not at All — Extremely

How **RELATED TO** and **CONTINGENT ON** your messages were the chat bot's responses?
(i.e. were the chat bot's responses related to what you said?)

○ ○ ○ ○ ○ ○ ○

Not at All — Extremely

How **EMPATHETIC** was the chat bot?
(i.e. did it respond in a **supportive** and **emotionally appropriate** way)?

○ ○ ○ ○ ○ ○ ○

Not at All — Extremely

Any other thoughts about the chat in general?

Figure 4.2: Screenshots of our Interactive Evaluation Platform (available at `https://neural.chat`): (a) chat window (left) and first part of the evaluation form (right); (b) second part of the evaluation form (to show all evaluation questions asked).

4.1.4.3 *Novel metrics and self-play*

Inspired by real-world human interactions, we introduce novel metrics to capture the morphology of a conversation, i.e., how the users' responses progress over time and how the bot's responses interact with them. We propose a hybrid combination of these metrics, M_H, that is optimized to predict conversation quality on human data. We then apply M_H to self-play, i.e., the trajectory of bot-generated responses, and investigate how it relates to human ratings of conversation quality.

4.1.5 *Details about implicit metrics*

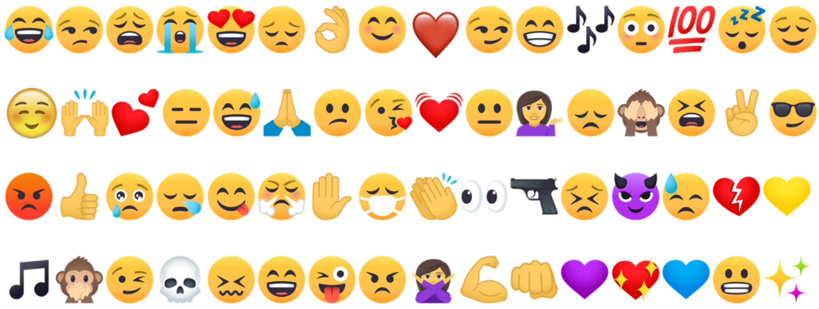


Figure 4.3: 64 most frequent emojis in the Twitter corpus, in order of frequency. DeepMoji (Felbo et al., 2017) predicts the probability of each emoji given an utterance, and this vector is used for calculating emotion embeddings.

Sentiment metrics To approximate emotional tone of an utterance, we use a state-of-the-art sentiment detector called *DeepMoji*, which was trained on a large Twitter corpus to predict the emojis used in tweets (Felbo et al., 2017). Transfer learning from this model to other tasks showed that it was able to significantly outperform state-of-the-art classifiers on a series of sentiment, irony, and sarcasm benchmarks, and a detailed analysis of the models predictions reveals a nuanced ability to distinguish between the tone of slightly different phrases. DeepMoji outputs an emotion embedding: a probability distribution over 64 most-frequently used emojis, shown in Figure 4.3. To estimate the *Sentiment Coherence* between user's query and generated samples, we calculate the cosine similarity between their emotion embeddings. We define a set of weights over the 64 emojis and calculate the weighted sum over an emotion embedding vector to derive a *Sentiment* score which is higher for positive sentiment and lower for negative sentiment; the weights are shown in Figure 4.4. We define *Sentiment Transition* as the change between user's *Sentiment* before and after a bot response. Additionally, *Sentiment Min-Max* is defined by the slope of change between min and max *Sentiment* in user utterances over the course of a conversation. This was designed to measure whether the peak positive sentiment occurred later in the conversation than the peak negative sentiment; we reasoned that sentiment should improve over the course of the conversation. Since humour can be used to create solidarity (Hay, 2000), we count the number of 'ha's in the user response as a proxy for *Laughter*. The combination of these metrics provides a snapshot of the trajectory of sentiment in a conversation and quantifies if the bot is able to elicit positive emotions in the user.

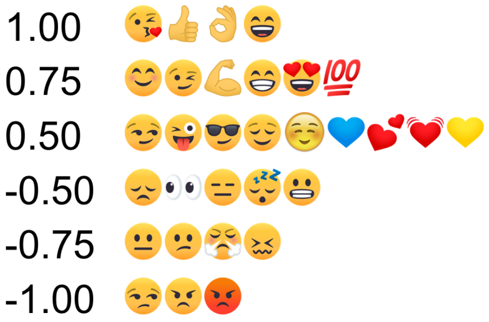


Figure 4.4: Assigned weights used for reducing the 64-dimensional emotion embedding into a *Sentiment* score.

Semantic metrics Language style matching is a strong predictor of relationship stability (Ireland et al., 2011) and social cohesiveness (Gonzales et al., 2010); thus, we introduce metrics to capture lexical similarity. We use *Infersent*, a state-of-the-art sentence-embedding model to encode the user and bot responses into a 4096-dimensional embedding space (Conneau et al., 2017). Infersent was trained to distinguish if two sentences are supporting, contradicting, or have a neutral relationship. We estimate *Semantic Similarity* by calculating the cosine similarity between the infersent embedding of the user's query and the generated bot sample. Additionally, we use the classic Word2Vec embeddings trained on Google News Corpus along with average, extrema, and greedy aggregation methods similar to Section 4.1.4.1 to derive *Average Word Coherence*, *Extrema Word Coherence*, and *Greedy Word Coherence* between user and bot responses.

Engagement metrics Asking questions is an important active listening skill which is linked to conversation management, attentiveness, and responsiveness (Bodie et al., 2012). Therefore, we define *Question Score*, to which we add 0.5 if the utterance contains a question word (*how, what, where, why, when, who*), and an additional 0.5 if it contains a question mark. We also introduce *# Words* as a proxy for user engagement that counts the number of words in their response. Based on prior work (Zhou et al., 2018b), we use the number of turns in the conversation as an indicator of the quality of the bot's performance. We also compute the number of words in the user's response, which we refer to as the *words elicited*.

Hybrid metric (M_H) We combine the aforementioned metrics (M_i) using linear regression, and optimize their coefficients (λ_i) to best predict human judgment of interactive conversation quality: $M_H =$

$\sum \lambda_i * M_i + M_0$. We use a leave-bot-out scenario where we isolate all the human conversations with one of the dialog models, χ_j, as the hold-out test set. We train the $\lambda_{i,j}$ on the remaining quality ratings. We found that the learned λ_is were stable across the training folds, only exhibiting small variations. Other researchers are encouraged to use our learned coefficients directly or adjust them according to their own interactive human evaluation dataset.

Self-play as an approximation for interactive evaluation Since interactive human evaluation is costly, we propose a *self-play* scenario where the dialog system talks to itself, i.e. the bot generated responses are fed back into it as the next turn input. For each model χ_j, we generate 100 random conversations, fixed at 10 turns. The self-play trajectories created using model χ_j are treated as the hold-out set. Therefore, the trained $\lambda_{i,j}$ values based on all conversations except for the ones with χ_j are used to calculate M_H on each generated bot-bot conversation trajectory for χ_j. The estimated M_H values are averaged across conversation samples for χ_j. This value is used for comparison against the ground-truth interactive quality ratings aggregated on a the bot-level.

4.1.6 *Experiments*

4.1.6.1 *Datasets*

A common source of data for open-domain dialog systems is movie scripts, among which the Cornell dataset (Danescu-Niculescu-Mizil and Lee, 2011) is the largest and most commonly used. Therefore, we use it to benchmark against previous state-of-the-art results (Park et al., 2018). Its median conversation length is 3 utterances and the conversations are strictly between pairs of speakers. Recognizing that movie lines have limited conversation diversity, we also built a new corpus, Reddit. Between the many different subreddits available, the conversations vastly differ on topic, language style, and participation patterns. We select the Casual Conversations forum (`https://www.reddit.com/r/CasualConversation`), a community of $607K$ conversationalists discussing a variety of topics. We collect a dataset of $109K$ conversations of at least 3 turns with the median conversation containing 7 utterances from conversational exchanges on the platform in 2018. This Reddit dataset is available at `https://affect.media.mit.edu/neural_chat/datasets` for public use.

4.1.6.2 *Interactive human evaluation results*

Table 4.1 (in Section 4.1.1) illustrates how EI regularization produces a higher quality conversation when compared to baseline. Rather than cherry-picking results, we make all of the bots evaluated in the study available at `https://neural.chat/BRFZACDCOA/` for readers to assess interactively.

		Cornell		Reddit	
Model	Metric	Baseline	EI	Baseline	EI
HRED	quality	2.182 ± 0.305	**2.347** ± 0.313	2.527 ± 0.310	**2.714** ± 0.299
	fluency	3.909 ± 0.387	**4.000** ± 0.381	4.436 ± 0.349	**4.786** ± 0.316
	diversity	**2.836** ± 0.374	2.735 ± 0.380	3.418 ± 0.386	**3.554** ± 0.372
	contingency	2.200 ± 0.291	**2.469** ± 0.336	2.382 ± 0.288	**2.536** ± 0.322
	empathy	**2.673** ± 0.352	2.490 ± 0.350	3.018 ± 0.329	**3.107** ± 0.337
VHRED	quality	2.022 ± 0.309	**2.333** ± 0.252	2.694 ± 0.392	**2.864** ± 0.341
	fluency	3.109 ± 0.351	**3.949** ± 0.396	4.250 ± 0.496	**4.477** ± 0.402
	diversity	3.565 ± 0.442	**4.385** ± 0.371	**5.00** ± 0.468	4.705 ± 0.353
	contingency	2.261 ± 0.287	**2.487** ± 0.346	2.472 ± 0.362	**2.773** ± 0.370
	empathy	**2.739** ± 0.374	2.564 ± 0.367	3.000 ± 0.393	**3.341** ± 0.385
VHCR	quality	2.132 ± 0.247	**2.548** ± 0.380	2.615 ± 0.350	**2.692** ± 0.298
	fluency	2.679 ± 0.306	**3.976** ± 0.380	3.923 ± 0.433	**4.308** ± 0.395
	diversity	3.755 ± 0.340	**4.238** ± 0.421	**4.436** ± 0.455	4.231 ± 0.382
	contingency	2.189 ± 0.270	**2.571** ± 0.356	2.077 ± 0.298	**2.692** ± 0.354
	empathy	2.340 ± 0.316	**2.714** ± 0.368	2.974 ± 0.434	**3.288** ± 0.379

Table 4.2: Mean ratings (from humans) for Baseline and EI (Emotion+Infersent) models for HRED, VHRED, and VHCR architectures with 90% confidence intervals. For 3-factor ANOVA results, see Section 4.1.6.2.

Overall, N=566 ratings were captured. Table 4.2 summarizes human ratings of baseline and EI models obtained via interactive evaluation. We ran a 3-factor ANOVA on the sum of user scores, where the independent variables are model architecture (HRED, VHRED, VHCR), EI regularization (Baseline, EI), and dataset (Cornell, Reddit). We found a significant main effect of EI regularization and dataset, but no significant difference between the three types of hierarchical models. We found that adding emotion and infersent (EI) regularization to baseline models improved the interactive chat experience significantly, $F(554, 1) = 9.016, p = .003$. Further, the models trained on the Reddit dataset performed significantly better, $F(554, 1) = 30.796, p < .001$. This finding validates the hypothesis that distilling information about topic and tone into the top level of the hierarchy is useful for good conversation, and suggests that the Reddit dataset could provide more realistic training for open-domain dialog and be valuable to the community.

4.1.6.3 Traditional metrics results

		Cornell					Reddit				
Model	Version	PPL	KL	Avg	Ext	Grd	PPL	KL	Avg	Ext	Grd
HRED	baseline	52.311	-	.471	.329	.331	41.730	-	.649	.394	.474
	EI	**47.636**	-	**.560**	**.383**	**.400**	**41.245**	-	**.651**	**.398**	**.482**
VHRED	baseline	**49.414**	.264	.539	.352	**.395**	36.240	**.188**	.635	.383	.464
	EI	50.526	**.517**	**.545**	**.355**	.394	**35.510**	.167	**.636**	**.392**	**.465**
VHCR	baseline	61.000	**.562**	.532	.345	.382	**36.736**	**.267**	.619	.371	.448
	EI	**49.243**	.475	**.588**	**.369**	**.444**	37.198	.231	**.639**	**.394**	**.469**

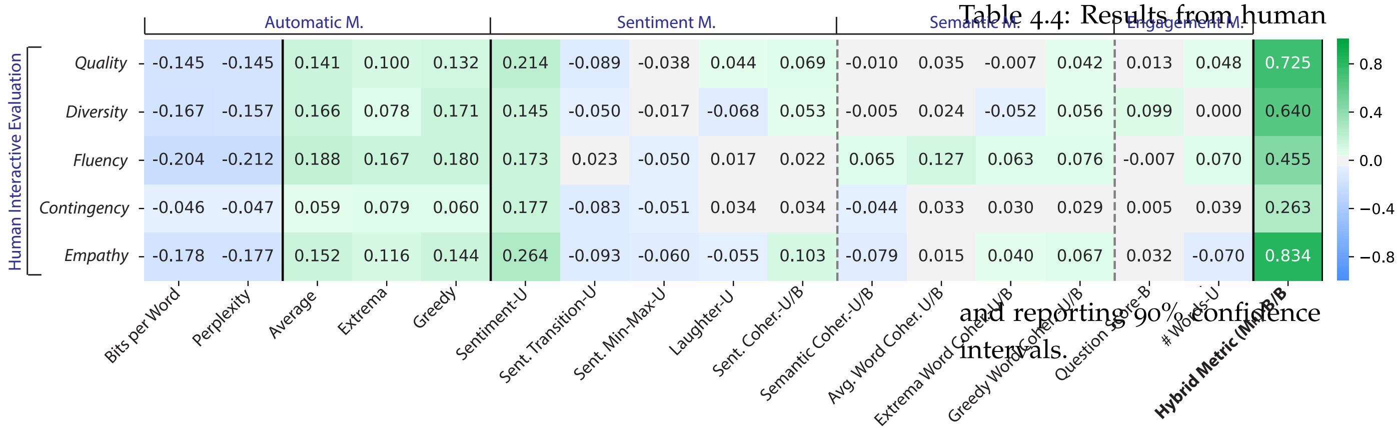
Table 4.3: Results of automatic traditional metrics for 1-turn responses of models per context of baseline and EI (Emotion + Infersent) models. PPL: perplexity, KL: KL divergence, Avg: Average, Ext: Extrema, Grd: Greedy

Automatic metrics Several prior works have focused on ensuring that the variational KL term remains high in order to improve model quality (e.g. (Shen et al., 2018; Park et al., 2018)). However, we observe there is no consistency between human quality rating and KL (Table 4.3). Thus, it is not evident that KL captures human judgements of dialog quality. Even perplexity (a transformation of the cross-entropy loss used to train our models) falls short of capturing human quality judgments, underscoring the difficulty in effectively training good language models. We find embedding metrics show more promise in preserving the order of human quality ratings, but have only weak correlation with human ratings. We present evidence for our novel hybrid metric being a much stronger alternative.

Human one-turn evaluation As shown in Table 4.4, while single-turn human evaluation suggests EI regularization is effective due to a higher number of win judgment, the results are noisy and difficult to interpret due to large confidence intervals and a high percentage of ties. The median inter-annotator agreement measured pairwise through Cohen's kappa (Fleiss et al., 1969) for our human evaluation was only 0.176 and 0.120 for CORNELL and REDDIT respectively. This level of annotator agreement is lower than the median Cohen's kappa of previous work (Liu et al., 2016) and explains the larger confidence intervals. Even after removing ambiguous examples (i.e. where equal number of annotators select each response as being better), large annotation variation persists. This may be due to subjective interpretations and ambiguity arising from different interpretations of <unknown> tokens or the short length of contexts in the CORNELL corpus (e.g. median length of conversation of 3 utterances). These findings further highlight the importance of an interactive evaluation as opposed to limited single-turn responses.

	Cornell			Reddit		
Model	Wins %	Losses %	Ties %	Wins %	Losses %	Ties %
HRED-EI	**40.8** ± 4.9	24.5 ± 4.9	34.8 ± 9.2	**31.3** ± 5.2	29.5 ± 6.6	39.3 ± 10.7
VHRED-EI	**36.9** ± 4.7	36.6 ± 5.6	26.6 ± 6.9	**39.0** ± 7.0	34.0 ± 5.3	27.0 ± 8.9
VHCR-EI	**33.0** ± 6.1	29.0 ± 5.4	38.0 ± 10.1	**33.7** ± 7.9	27.3 ± 3.3	39.0 ± 8.6

Table 4.4: Results from human and reporting 90% confidence intervals.



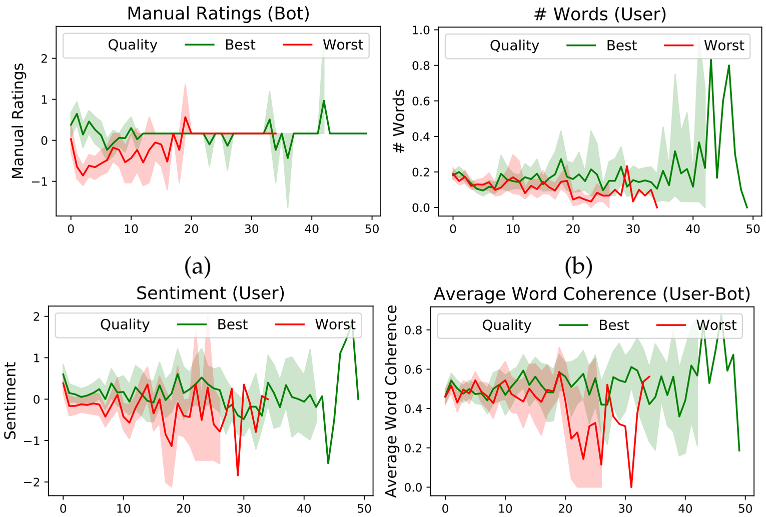
	Bits per Word	Perplexity	Average	Extrema	Greedy	Sentiment-U	Sent. Transition-U	Sent. Min-Max-U	Laughter-U	Sent. Coher.-U/B	Semantic Coher.-U/B	Avg. Word Coher. U/B	Extrema Word Coher. U/B	Greedy Word Coher. U/B	Question Score-B	# Words-U	**Hybrid Metric (M_H) B/B**
Quality	-0.145	-0.145	0.141	0.100	0.132	0.214	-0.089	-0.038	0.044	0.069	-0.010	0.035	-0.007	0.042	0.013	0.048	0.725
Diversity	-0.167	-0.157	0.166	0.078	0.171	0.145	-0.050	-0.017	-0.068	0.053	-0.005	0.024	-0.052	0.056	0.099	0.000	0.640
Fluency	-0.204	-0.212	0.188	0.167	0.180	0.173	0.023	-0.050	0.017	0.022	0.065	0.127	0.063	0.076	-0.007	0.070	0.455
Contingency	-0.046	-0.047	0.059	0.079	0.060	0.177	-0.083	-0.051	0.034	0.034	-0.044	0.033	0.030	0.029	0.005	0.039	0.263
Empathy	-0.178	-0.177	0.152	0.116	0.144	0.264	-0.093	-0.060	-0.055	0.103	-0.079	0.015	0.040	0.067	0.032	-0.070	0.834

Figure 4.5: Correlations between five human metrics and automated metrics (-U: Calculated on user response, -B: bot response, -U/B: between user and bot response, -B/B: between consecutive bot utterances). **Hybrid Metric $\mathbf{M_H}$ -B/B**, our novel self-play based metric, has a higher correlation across all human metrics than any other metric proposed to-date.

4.1.6.4 Novel metrics applied to human data and self-play

Figure 4.5 summarizes the relationships between interactive human ratings and the automated metrics. We observe that our sentiment metric applied to human data on its own has higher correlation with interactive human ratings than the commonly used metrics such as perplexity and embedding distance metrics. Most importantly, our novel hybrid metric, M_H, applied to self-play aggregated on the model-level is strongly correlated with all human ratings ($r > .7$), while previous metrics achieved $r < .5$. This is a significant finding, suggesting that even without running interactive human evaluation, we can automatically approximate it through self-play. This metric is agnostic to the training set and model type and can be calculated on the trajectory of self-play utterances for any chatbot, regardless of its architecture. One interpretation is that the self-play framework keeps the conversation within the training set distribution, and the model is less likely to produce <unknown> tokens. Therefore, M_H and its sub-components have meaningful values for the generated responses and can be useful for quality approximation.

Though we expect that the hybrid nature of M_H makes it less exploitable, optimizing for its sub-components in isolation through a self-play scenario should be avoided. Differently from human interaction, maintaining extreme similarity in sentiment or semantics or just



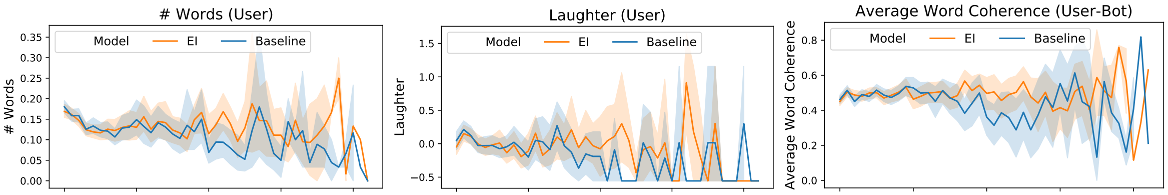
(a) (b)

(c) (d)

Figure 4.6: One hundred highest vs. lowest quality conversation trajectories; lines: mean, shaded area: 90% confidence intervals, x-axis: conversation turns. (a) Timing of upvote/downvote ratings: A bad first impression impedes overall rating. (b) Participants talk longer and use more words in conversations rated higher. (c) High-quality conversations elicit more positive user sentiment. (d) High-quality conversations are more semantically similar as measured by average word coherence between user query and bot responses. Note that users tend to leave the conversation after strong negative sentiment or semantic dissimilarity.

asking questions in self-play conversation trajectories could backfire by reducing the diversity of generated responses.

We examine how the novel psychologically-inspired metrics relate to the trajectories of the 100 best and 100 worst quality conversations. This is only feasible with interactive evaluation. As shown in Figure 4.6, we observe that appropriate sentiment, coherent semantics, and engaging users are indispensable to attaining high quality ratings in multi-turn interaction. Comparing EI and baseline conditions, we see a replication of these trends (Figure 4.7). For example, EI elicits longer responses from users (greater engagement), with more laughter and higher semantic coherence.



(a) (b) (c)

Figure 4.7: EI vs. baseline conversation trajectories; lines: mean, shaded area: 90% confidence intervals, x-axis: conversation turns. (a) EI elicits longer responses from users, suggesting that they are more engaged compared to baseline. (b) EI evokes more laughter from users compared to baseline. (c) EI has higher semantic coherence as measured by average word coherence. The same pattern applies to greedy and extrema word coherence.

4.1.7 *Conclusions*

A major obstacle in open-domain dialog generation is the predominant optimization of an objective function that does not map out to human judgment of conversation quality in a naturalistic chat. In this section, we have argued that it is necessary to go beyond single-turn evaluation by investigating the strengths of interactive evaluation and highlighting blind-spots of traditional one-turn evaluation methods. To alleviate this problem, we have combined interactive human data with psychologically-motivated measures and introduced a novel hybrid metric. Using this metric in a self-play framework provides results that are strongly correlated with human judgment of chatbot empathy (r>.8) and quality (r>.7). Additionally, we have demonstrated a significant improvement to several hierarchical seq2seq generative models using regularization of the utterance level of the hierarchy with knowledge distillation. Finally, we have open-sourced the platform together with a new REDDIT dataset.

4.2 Way off-policy batch reinforcement learning

Given the human interaction data collected in the previous section, we would like to use reinforcement learning (RL) to learn to improve our models based on human feedback. However, most deep RL systems are not able to learn effectively from off-policy data, especially if they cannot explore online in the environment. These are critical shortcomings for applying RL to real-world problems where collecting data is expensive, and models must be tested offline before being deployed to interact with the environment – e.g. systems that learn from human interaction. Thus, we develop a novel class of off-policy batch RL algorithms, which use KL-control to penalize divergence from a pre-trained prior model of probable actions. This KL-constraint reduces extrapolation error, enabling effective offline learning, without exploration, from a fixed batch of data. We also use dropout-based uncertainty estimates to lower bound the target Q-values as a more efficient alternative to Double Q-Learning. This Way Off-Policy (WOP) algorithm is tested on both traditional RL tasks from OpenAI Gym, and on the problem of open-domain dialog generation; a challenging reinforcement learning problem with a 20,000-dimensional action space. WOP allows for the extraction of multiple different reward functions post-hoc from the collected human interaction data, and can learn effectively from all of these. We test the real-world generalization by deploying dialog models live to converse with humans in an open-domain setting, and demonstrate that WOP achieves significant improvements over state-of-the-art prior methods in batch deep RL.

4.2.1 *Background*

In order to scale deep reinforcement learning (RL) to safety-critical, real-world domains, two abilities are needed. First, since collecting real-world interaction data can be expensive and time-consuming, algorithms must be able to learn from off-policy data no matter how it was generated, or how little correlation between the data distribution and the current policy. Second, it is often necessary to carefully test a policy before deploying it to the real world; for example, to ensure its behavior is safe and appropriate for humans. Thus, the algorithm must be able to learn offline first, from a static batch of data, without the ability to explore.

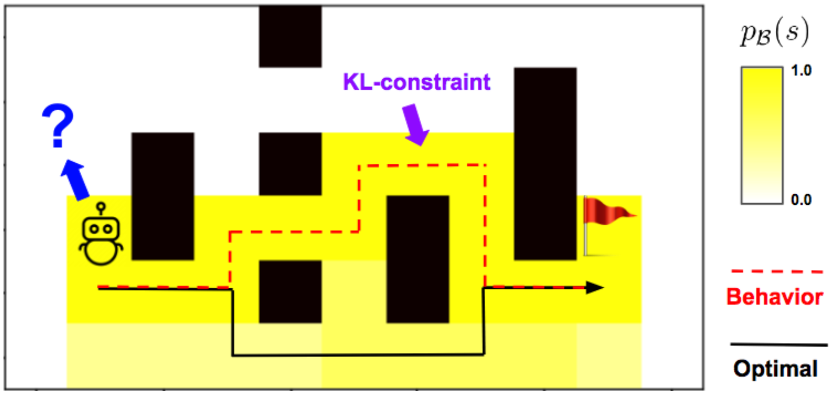


Figure 4.8: In this example batch RL problem, the robot's goal is to travel the minimum distance around the black walls to get to the red flag. A trained behavior policy generated the batch data; the probability of each of the states appearing in the batch, $p_{\mathcal{B}}(s)$, is in yellow (white locations are not contained in the batch). If the offline RL policy estimates the value of going *up* or *left* from the start position is high, it will have no way to refine this estimate using the batch data, or learn a good policy in this region of state space. The KL-constraint ensures that the RL policy will stay within the support of the batch data. However, the behavior policy is suboptimal, so using behavior cloning to directly imitate the batch data will result in suboptimal return. Instead, the KL-constrained model can learn to find the optimal policy, which is within the support of the batch.

This off-policy, *batch reinforcement learning* (BRL) setting represents a challenging RL problem. Most deep RL algorithms fail to learn from data that is not heavily correlated with the current policy (Fujimoto et al., 2018b). Even models based on off-policy algorithms like Q-learning fail to learn in the offline, batch setting, when the model is not able to explore. If the batch data is not sufficient to cover the state-action space, BRL models can suffer from *extrapolation error*, learning unrealistic value estimates of state-action pairs not contained in the batch (Fujimoto et al., 2018b). It can be impossible to correct for extrapolation error when there is a mismatch in the distribution of state-actions pairs in the batch data, and the distribution induced by the learned policy. For example, if the policy learns to select actions which are not contained in the batch, it cannot learn a reasonable value function for those actions. Figure 4.8 illustrates this concept, where the batch only covers a subset of possible policies. Extrapolation error is particularly problematic in high-dimensional state and action spaces (such as those inherent in language generation).

We propose to resolve these issues by leveraging a pre-trained generative model of the state-action space, $p(a|s)$, trained on known sequences of interaction data. While training with RL, we penalize divergence from this prior model with different forms of KL-control. This technique ensures that the RL model learns a policy that stays close the state-action distribution of the batch, combating extrapolation error. We also propose using dropout to obtain uncertainty estimates of the target Q-values, and use this lower bound to alleviate overestimation bias. We benchmark against a discrete adaptation of Batch Constrained Q (BCQ) (Fujimoto et al., 2018b), a recently proposed BRL algorithm for continuous domains, and show that our Way Off-Policy algorithm achieves superior performance in both a traditional RL domain, as well as in a challenging, under-explored, real-world reinforcement learning problem: using implicitly expressed human reactions in chat to improve open-domain dialog systems.

When a machine learning system interacts with humans, ideally we would like to learn about the humans' preferences in order to improve the performance of the system. Yet having humans manually indicate their preferences through explicit means like pressing a button (e.g. (Christiano et al., 2017)) or submitting a feedback report, does not scale. Instead, we would like to be able to use humans' implicit reactions, such as the sentiment they express, or the length of the conversation, in order to improve the policy.

Applying off-policy batch RL to language generation is challenging because the number of potential combinations of words and sentences leads to a combinatorial explosion in the size of the state space. The action space—the set of frequent vocabulary words in the English language—is 20,000-dimensional. This compounds extrapolation error, making BRL even more difficult. However, when learning from human interactions in the wild, it is crucial to be able to learn offline and test the policy before deploying it, lest it learn inappropriate behaviors (e.g. (Horton, 2016)).

To support this work, we developed an interactive online platform that allows humans to chat with deep neural network dialog models running on GPU; the BRL models trained for this study are available live at `https://neural.chat/rl`. Through this platform we collected human responses to a set of over 40 different dialog models over the course of several months. Using our Way Off-Policy algorithm, we are able to effectively learn from this batch of data, in spite of the fact that it was generated with a vastly different set of model architectures, which were trained on different datasets. Further, we use the batch to learn from many different reward functions designed post-hoc to extract implicit human preferences, something that is only possible with effective off-policy BRL.

In summary, the contributions of this section are:

- A novel algorithm, Way Off-Policy learning, which is the first to propose using KL-control from a pre-trained prior model as a way to reduce extrapolation error in batch RL.
- Experiments showing the effectiveness of WOP above strong baselines based on prior work (e.g. Fujimoto et al. (2018b)), on both traditional RL tasks and on the challenging problem of open-domain dialog generation.
- A set of novel conversation rewards based on how human preferences are implicitly expressed in text. We are the first work to learn from implicit signals in conversation offline using batch RL.

4.2.2 Related Work

The approach we propose is based on KL-control, a branch of stochastic optimal control (SOC) (Stengel, 1986) where the Kullback-Leibler (KL) divergence from some distribution is used to regularize an RL policy (e.g. (Abdolmaleki et al., 2018; Kappen et al., 2012; Rawlik et al., 2012; Todorov, 2007)). Well-known examples include Trust Region Policy Optimization (TRPO) (Schulman et al., 2015), and use conservative, KL-regularized policy updates to restrict the RL algorithm to stay close to its own prior policy (e.g. (Haarnoja et al., 2018; Kakade, 2002; Peters et al., 2010; Rawlik et al., 2012)). KL-control can also be applied to entropy maximization (e.g. (Ziebart et al., 2008; Nachum et al., 2017; Haarnoja et al., 2017)); for example, G-learning penalizes KL-divergence from a simple uniform distribution in order to cope with overestimation of Q-values (Fox et al., 2016). Soft Q-learning motivates using a Boltzmann distribution in the value function as a way of performing maximum entropy RL (Haarnoja et al., 2017). KL-control has also been used to improve transfer learning between maximum likelihood estimation (MLE) training on data, and training with RL (Jaques et al., 2017a). To the best of our knowledge, our work is the first to propose penalizing KL-divergence from a learned prior model of the state-action space as a way to improve offline batch RL.

Other strategies to improve off-policy learning have been proposed, but differ from this work in key respects. Many focus on scenarios where the policy is able to explore and collect more data (e.g. (Degris et al., 2012; Riedmiller, 2005)); for example, when learning online from an outdated replay buffer (e.g. Munos et al. (2016)). In contrast, we learn entirely offline, from a fixed batch of data, without the ability to explore. Methods proposed for this setting have often not been used in conjunction with modern function approximation techniques (e.g. Thomas et al. (2015)). Many other works focus on off-policy policy *evaluation* (rather than policy learning), for example using importance sampling or model estimation (e.g. Farajtabar et al. (2018); Jiang and Li (2016); Precup (2000); Thomas and Brunskill (2016)).

In the deep BRL setting, Liu et al. (2019) have proposed a correction to policy gradients, Gelada and Bellemare (2019) have proposed covariance-shift methods, and Bhatt et al. (2019) have proposed normalized feature representations. Kumar et al. (2019) use maximum mean discrepancy to cope with extrapolation error in BRL, while Agarwal et al. (2019) use a Random Ensemble Mixture (REM) Q-network. Most similar to our work is Batch Constrained Q-learning (BCQ) (Fujimoto et al., 2018b), which tackles off-policy BRL in continuous action domains by training a generative model of the batch, $p(a|s)$, sampling from this model, and selecting the best action based on a Q-estimate. Unlike our approach, this does not integrate information about the distribution $p(a|s)$ directly into the policy, or allow the model to learn when to strategically deviate from the prior in order to obtain more reward.

We propose using dropout to approximate model uncertainty of the target Q-network. The idea of using dropout to estimate uncertainty in neural networks was first proposed by Gal and Ghahramani (2016). Different forms of uncertainty estimates have been used in RL (e.g. (Kahn et al., 2017; Osband et al., 2016)); for example, Bayesian uncertainty estimates have been proposed as an alternative to double DQN (Azizzadenesheli et al., 2018).

4.2.3 *RL for language generation*

Improving dialog systems with RL has largely been restricted to task-oriented dialog systems, which have a limited number of task-specific actions (e.g. (Fatemi et al., 2016; Gašić et al., 2011; Liu and Lane, 2017; Liu et al., 2018; Su et al., 2017)). These approaches may incorporate human input, usually through explicit, manual feedback (e.g. (Shah et al., 2018)), but sometimes with more implicit signals, such as the user interrupting the system or starting over (Shi and Yu, 2018). Efforts to expand RL to the open-domain dialog setting, such as those of Li et al. (2016c, 2017, 2018b), are less numerous, and do not involve learning from human feedback. Even in the open-domain setting, authors may choose to use a highly restricted action space; for example, using RL to choose which scripted or MLE dialog model to invoke to answer a user's query (Serban et al., 2017a).

Since the posting of the preprint of this paper, Ziegler et al. (2019) have used explicit human feedback to improve the summarization and text continuation performance of a large-scale language model. Although they do not study dialog or the batch RL setting (instead learning online from a trained model of human feedback), they do make use of our proposal to penalize KL-divergence from a pre-trained language model, and find that this is important to achieving good performance.

Although implicit signals such as sentiment (Hancock et al., 2019) and conversation length (Zhou et al., 2018b) have been used in MLE systems, the idea of using such signals as a reward for RL is relatively unexplored. Shin et al. (2019) use on-policy learning in conjunction with a user-sentiment approximator to improve a seq2seq model, but are unable to learn directly from user feedback. To the best of our knowledge, we are the first to use batch RL to train open-domain dialog models on implicit cues gained from real human interactions.

4.2.4 *Methods*

We employ typical RL notation in which s_t represents the environment state at time t, the agent takes action a_t according to its policy $\pi(a_t|s_t)$, and receives a reward $r(s_t, a_t)$. The agent's goal is to maximize reward over an episode trajectory τ, with a discount factor of γ applied to future rewards. Q-learning methods learn an action-value estimate of the total expected discounted future reward,

$Q_\pi(a_t, s_t) = \mathbb{E}_\pi[\sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})]$, through iterative updates based on the Bellman equation:

$$Q_{\theta_\pi}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p(\cdot|s_t, a_t)}[\max_{a_{t+1}} Q_{\theta_T}(s_{t+1}, a_{t+1})] \tag{4.6}$$

In deep Q-learning (Mnih et al., 2013), a Q-network approximates $Q_{\theta_\pi}(s_t, a_t)$ and drives the policy π. A second target Q-network approximates the expected reward from the next state, $Q_{\theta_T}(s_{t+1}, a_{t+1})$.

4.2.4.1 *Batch RL and extrapolation error*

In batch RL, we are given a fixed batch of data $\mathcal{B}$, and assume that no further interaction with the environment is possible. To train Q_{θ_π}, we sample $(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{B}$, and update the weights of the Q-network to approximate Eq. 4.6. Because Q-learning is an off-policy algorithm, in principle it should be able to learn from data collected by any behavior policy. However, extrapolation error can occur if the BRL policy learns to favour a state-action pair (s, a) that is unlikely, or not contained, in the batch data. In this case, the estimate $Q(s', \pi(s'))$ can be arbitrarily bad (Fujimoto et al., 2018b). Such errors can then accumulate through the Bellman backup operator (Kumar et al., 2019). Experiments from Fujimoto et al. (2018b) show that extrapolation error can be highly detrimental to learning off-policy in BRL.

These problems are compounded by the fact that algorithms based on the Bellman operator are inherently optimistic in the face of uncertainty. When value estimates for some region of the state-action space are noisy (because too few experience samples have been used to refine them), the maximum operation in Eq. 4.6 will lead to an overestimation of expected future reward. In a normal RL setting, this overestimation bias drives the model to explore areas of the state-action space for which the value estimates have the highest variance, thus enabling it to refine them; in essence, creating a built-in drive to explore. However, in a batch setting where exploration is not possible, the model is instead driven to value parts of the state-action space for which it has little to no data to learn a good policy (see Figure **??**).

4.2.4.2 *Dropout for uncertainty estimation of Target Q-values*

The overestimation of Q-values in the BRL setting necessitates other methods for estimating the future reward via the Target Q-network. Clipped Double Q-learning (Fujimoto et al., 2018a) maintains two independent pairs of Q-networks, and taking the minimum of their estimates of future reward. This approach is computationally expensive and memory intensive.

Instead, we obtain a distribution over predictions from a single target Q-network trained with dropout, and take the lower bound of these to reduce overestimation bias. It has been shown that dropout approximates Bayesian uncertainty for neural networks, by assuming the weights of the network are drawn from a Gaussian prior, $W \sim N(0, I)$, and using variational inference to estimate the posterior distribution $p(W|X, Y)$ (Gal and Ghahramani, 2016). We perform dropout during both training and inference before each weight layer, and approximate the posterior such that the dropout distribution q^W is a mixture of Gaussians, and $D_{KL}[q^W||p(W|X, Y)]$ is minimized. Given the target Q-network Q_{θ_T}, we compute $Q(a_{t+1}, s_{t+1})$ using a Monte Carlo (MC) estimate of the lower-bound of $Q_{\theta_T}(a_{t+1}, s_{t+1})$ by running M stochastic forward passes of the network, each with a new dropout mask $d_i \sim q^W$:

$$Q(a_{t+1}, s_{t+1}) = \min_{i=1...M} [Q_{\theta_T}(a_{t+1}, s_{t+1}; d_i)] \tag{4.7}$$

Using the minimum operator penalizes high variance estimates and leads the algorithm to be pessimistic in the face of uncertainty, rather than optimistic. Such a bias will push the model to favour actions that lead to states well covered by the batch data (Fujimoto et al., 2018b).

4.2.4.3 *Discrete Batch Constrained Q*

Batch Constrained Q-learning (BCQ) (Fujimoto et al., 2018b) proposes to address the BRL problem by constraining the actions of the Q-network to be close to the data contained within the batch. This is accomplished by learning a generative model of the batch, $G_w = p(a|s)$, and sampling from this model during learning and inference. Because BCQ is designed for continuous action domains, it applies a learned perturbation model $\xi(s, a; \Phi)$ which is allowed to alter the action within the range $[-\Phi, \Phi]$. BCQ learns Q-estimates that incorporate the perturbation model, $Q_\theta(s, a + \xi(s, a; \Phi))$. To act,

n possible actions are sampled from the generative model, $\{a_i \sim G_w(s)\}_{i=1}^n$, perturbed, and the action with the maximum Q-value is selected, giving the BCQ policy:

$$\pi_{BCQ}(s) = \underset{a_i+\xi(s,a_i;\Phi)}{\arg\max}\ Q_\theta(s, a_i + \xi(s, a_i; \Phi)) \tag{4.8}$$

We propose an adaptation of BCQ to discrete action spaces (*DBCQ*) which does not use a continuous perturbation model. Since BCQ relies on Double Clipped Q-learning (Fujimoto et al., 2018a), here we use dropout-based uncertainty estimates as in Eq. 4.7. Thus, the DBCQ policy is:

$$\pi_{DBCQ}(s) = \underset{a_i \sim p(a|s)}{\arg\max}\, Q_{\theta_\pi}(s, a_i) \tag{4.9}$$

4.2.4.4 KL Control from pre-trained prior

Rather than simply sample from the prior, we would like the Q-learning algorithm to directly incorporate the prior into the policy. Thus, we use KL-control to penalize divergence between the prior $p(a|s)$, and the Q-network policy π_θ, while still maximizing reward. Given a trajectory of actions, $\tau = \{a_1, a_2, ...a_{t-1}\}$, let $q(\tau) = \prod_{t=1}^T \pi_\theta(a_t, s_t)$ be the policy of our Q-learning algorithm at the trajectory level. Similarly, let $p(\tau) = \prod_{t=1}^T p(a_t|s_t)$ be the prior distribution over the trajectory, and $r(\tau)$ be the return. We seek to maximize the following KL-regularized objective:

$$L(q) = \mathbb{E}_{q(\tau)}[r(\tau)]/c - D_{KL}[q(\tau)||p(\tau)] \tag{4.10}$$

Since $D_{KL}[q||p] = \sum_x q(x)(\log q(x) - \log p(x))$, we can see that this is equivalent to maximizing the following expected value function of the policy π_θ at the action level:

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi[\sum_{t'=t}^{T} r(s_{t'}, a_{t'})/c + \log p(a_{t'}|s_{t'}) - \log \pi(a_{t'}|s_{t'})] \tag{4.11}$$

The two terms we have introduced in Eq. 4.11 have clear motivations. The $p(a|s)$ term rewards the model for choosing actions that have high probability under the prior, biasing the model to state-action pairs that are realistic, and likely to be in the batch. The $-\log \pi(a|s)$ term is analogous to entropy regularization. Maintaining diversity in the action space through entropy regularization is important for generative models like dialog systems, which are known to collapse to an uninteresting, small number of repeated samples (Li et al., 2016b). Re-stating Eq. 4.11 as an entropy-regularized Q-function, we obtain:

$$Q(s_t, a_t) = \mathbb{E}_\pi[\sum_{t'=t}^{T} r(s_{t'}, a_{t'})/c + \log p(a_{t'}|s_{t'}) + \mathcal{H}(\cdot|s_{t'})] \tag{4.12}$$

One can derive a soft version of the entropy-regularized Q-function that uses a Boltzmann distribution to estimate future reward (Haarnoja et al., 2017). We refer to it as a Ψ-function following previous work (Jaques et al., 2017a), which derived this function as a generalization of the Ψ-learning proposed by (Rawlik et al., 2012). The optimal Ψ-function and policy are:

$$\Psi^*(s_t, a_t) = r(s_{t'}, a_{t'})/c + \log p(a_{t'}|s_{t'}) + \gamma \log \sum_{a'} \exp(\Psi^*(s', a')) \tag{4.13}$$

$$\pi^*_\Psi(a_t|s_t) = \exp(\Psi^*(s_t, a_t)) \tag{4.14}$$

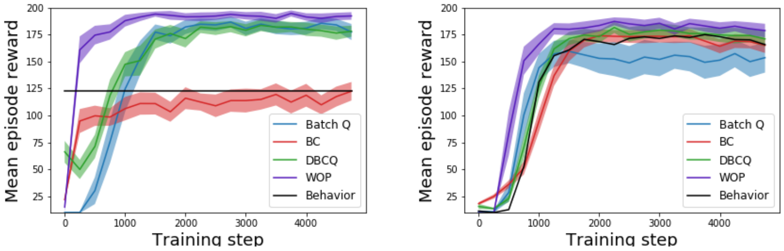
Because it avoids taking a hard max over noisy estimates, Ψ-learning leads to less overestimation of future reward (Abdolmaleki et al., 2018; Haarnoja et al., 2017). This improves learning through more stable temporal-difference (TD) updates. Thus, we argue it will be especially useful in the BRL setting for reducing optimism in the face of uncertainty. The Way Off-Policy (WOP) algorithm combines Monte Carlo (MC) target estimation, Ψ-learning, and KL-control from a pre-trained prior.

4.2.4.5 *Model averaging*

Finally, we explore the setting where the data in the batch may be generated from a large variety of different models M with different architectures, which each learn a different estimate of $p(a|s; M)$. We use this diversity to create a more robust prior by computing a weighted average of these models based on a normalized score $S(M)$ for each model. The score could be some measure of model quality, or simply the proportion of data in the batch that was generated with that model. Thus, we define $p_{MA}(a|s)$ as the model-averaged prior: $p_{MA}(a|s) = \sum_M S(M)p(a|s; M)$.

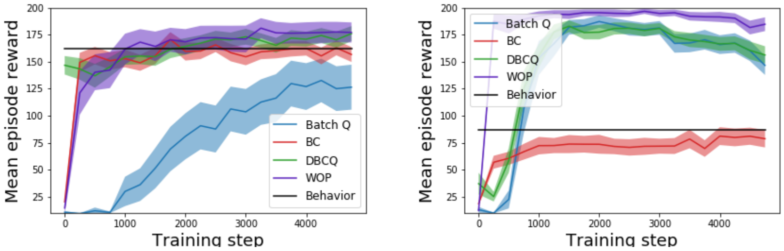
4.2.5 *Traditional RL experiments*

To demonstrate the effectiveness of these techniques, we conduct a series of experiments in traditional RL tasks using the OpenAI gym



(a) Full buffer

(b) Concurrent



(c) Expert demonstrator

(d) Noisy demonstrator

Figure 4.9: Comparison of batch RL algorithms for different offline learning conditions. WOP consistently exceeds the performance of Batch Q-learning, Behavioral Cloning (BC), DBCQ, and the Behavior policy used to generate the batch data. Error bars show 95% *CI* of the mean over 50 trials.

(Brockman et al., 2016). Here we show results for the *CartPole-v0* environment; more results are available in the Appendix. We first train an online Q-learning *Behavior* policy, and store all (s, a, r, s') experience samples into a replay buffer. We use this buffer to train a prior model of $p(a|s)$ using a Variational Auto-encoder (VAE) (details in Appendix). This model is used as a part of both the DBCQ and WOP algorithms. We can use the prior for imitation learning, by sampling actions directly from $p(a|s)$ to obtain Behavioral Cloning (BC). We benchmark all of these techniques against vanilla Q-learning on the batch data (*Batch Q*).

We experiment with four different conditions which vary the quality of the Behavior policy and the replay buffer data: a) *Full buffer*: all experience samples experienced during online training are used for offline learning; b) *Concurrent*: the offline learning algorithms see a sliding window of experience samples in the same order that the online learner experienced them; c) *Expert demonstrator*: the buffer only contains experience generated by a fully trained online learner; and d) *Noisy demonstrator*: the online learner has a high probability of acting randomly ($\epsilon = 0.3$) and is thus a bad model of the optimal policy.

Figure 4.9 shows the results. Across conditions, we see that WOP is able to outperform Batch Q, imitation learning (BC), DBCQ, and the original behavior policy. As expected, Imitation learning (BC) underperforms other techniques when the batch contains noisy or inexpert experience samples. However, when the batch contains only expert trajectories, Batch Q fails to learn, because the batch does not cover the full state-action space well, increasing extrapolation error (as illustrated in Figure **??**). DBCQ matches or outperforms BC and Batch Q in all scenarios. However, because DBCQ acts by sampling from $p(a|s)$ as learned by the BC model, its performance suffers when the batch data is noisy or imperfect. In contrast, WOP is able to learn to trade-off staying close to the prior and obtaining higher reward, and consistently outperforms all other algorithms in this environment.

4.2.6 Batch RL for learning dialog from human feedback

Here, we tackle the problem of training an open-domain dialog model from human feedback. We consider human interaction to represent the 'environment'. The response of a human to the bot's utterance is used to compute a reward signal to train the model. The state is the conversation history, composed of a series of conversation turns or utterances, $u_{1...t}$, where each utterance is composed of vocabulary tokens. The model attempts to construct a response utterance $u_{t+1}^{\pi} = [a_1, a_2, ..., a_n]$ by iteratively choosing an action a_i as the next token. Applying RL to dialog generation is challenging due to the large state-action space. The number of tokens in the vocabulary of our pre-trained model is 20,000, making the action space very high-dimensional; this further compounds the problem of extrapolation error.

The state of the environment s_t constitutes all of the text in the conversation uttered so far, both by the bot and the human. The state has a hierarchical structure, marking its division into utterances, which are further divided into tokens. While the bot is constructing an utterance u_t^{π}, it chooses its next action a_{ti} and the state s_t is updated by appending a_i: $s_t = [s_t, a_i]$. During this phrase, it is straightforward to obtain a target Q-estimate of future reward using the model's estimated Q-values over its own next token in the utterance. However, at the last token of the bot's utterance, the estimated future reward must include the human's response u_t^h. Therefore, we append the human response into the conversation, $s_{t+1} = [s_{t-1}, u_t^{\pi}, u_t^h]$, feed this into the target Q-network, and use the estimated Q-values for the first token of the bot's next utterance. The combinatorial size of the state space also increases extrapolation error , Therefore, we begin by initializing the Q-networks with the weights of a pre-trained language model. We use the best models obtained from the experiments in Section 4.1, which provide a strong prior over the appropriate word to select. The code for the models and the server is available in open-source at `https://github.com/natashamjaques/neural_chat/tree/master/rl`. Using the server, we collected a batch of human interaction data containing 14232 pairs of user input and agent response. The batch data was used to train the RL models as described in Section 4.2.4. Here, we use the pre-trained language model to estimate $p(a|s)$. We recruited 90 Mechanical Turk workers to provide a total of 718 7-point Likert scale ratings of the bots' quality, fluency, diversity, contingency (relatedness), and empathy, after interacting with each bot for at least 3 turns. Participants also had the option to provide explicit feedback through upvoting or downvoting a particular utterance within the interface. We sum these manual votes to create an overall *votes* score. We note that using this platform to test our models "in the wild" with humans represents a more meaningful test of generalization than testing an RL model in the same limited (game) environment in which it was trained, since humans are not restricted in the text they can type as input to the model.

4.2.6.1 Learning from implicit human preferences

We would like to improve a dialog model's ability to engage in natural conversation with a human by learning from the signals implicit in the way that the human responds. Rather than having the human manually label good performance—which we show in this work does not scale—the agent should recognize informative cues within the user's responses, like sentiment, and the amount of time they spend chatting. Essentially, we want to create an agent that is intrinsically motivated to produce positive reactions in its human conversation partner. Therefore, we employ the sentiment, semantic, and engagement metrics proposed in Section 4.1.4.3 as our rewards. We test combinations of these rewards, including the Hybrid Metric proposed in the previous section, but eventually settled on a final reward computed using the following equation:

```
r = 0.157*question + 0.138*semantic_coherence + 0.153*laughter
+ 0.142*sentiment_transition + 0.142*sentiment + 0.148*words_elicited
+ 0.120*conversation_length.
```

Note that with the exception of the *question* reward, these rewards depend on eliciting positive responses from a human, and are thus difficult to optimize. However, the *question* reward is trivially exploitable, since it depends only on the bot's own output. Finally, it is important to point out that these rewards represent only an initial foray into designing good metrics of human enjoyment, and further experimentation will be needed to improve them.

4.2.7 *Dialog results*

Model type	**Quality**	**Fluent**	**Diverse**	**Related**	**Empathy**	**Total**	**Votes**	**Human reward**
DBCQ	1.64 ±.29	1.87 ±.34	**3.13 ±.58**	1.84 ±.34	2.09 ±.38	10.58 ±1.55	-.228	-.050
Batch Q	1.87 ±.30	2.36 ±.42	2.20 ±.41	1.91 ±.32	2.58 ±.47	11.91 ±1.58	-.163	-.005
Batch Q + MC	1.85 ±.39	2.46 ±.44	2.46 ±.52	1.98 ±.39	2.34 ±.49	11.07 ±1.82	-.068	.005
KL-control Q	2.38 ±.39	3.24 ±.47	3.42 ±.54	2.38 ±.45	2.56 ±.43	13.98 ±1.81	.016	.004
KL-control Ψ	2.33 ±.41	**3.73 ±.53**	2.82 ±.50	2.31 ±.44	**3.47 ±.50**	**14.67 ±1.82**	**.128**	**.061**
KL-control MA Ψ	**2.60 ±.43**	3.47 ±.42	3.00±.49	**2.49 ±.44**	2.89 ±.51	14.44 ±1.96	.127	.042

Table 4.5: Interactive human evaluation of techniques for off-policy batch RL. KL-control models significantly out-perform other techniques. Ratings are Likert scale, votes and human reward are z-scores.

To compare models, we not only look at human users' ratings and votes, but also consider the automatic signals detectable from the text itself. This implicit *human reward* metric aggregates the measures listed in items 1-4 in Section 4.2.6.1, and measures the ability to elicit positive responses from the human. Table 4.5 shows the results of the human evaluation, comparing WOP to ablations of itself, Batch Q, and DBCQ[2]. MC Target Q estimation leads to modest improvements in *votes* and *human reward*, but does not improve ratings. Conversely, model averaging (MA) leads to modest improvements in ratings without improving votes or reward. Using Ψ-learning improves all three. However, the most notable difference in performance comes from KL-control. The KL-control models show substantial gains over the baseline models across both ratings and human reward. We perform a one-way analysis of variance (ANOVA) comparing the KL-control models to the Batch Q baselines and DBCQ on the total human rating score, and find that the KL-control models are significantly better, $F(x) = 4.781, p < .05$. This validates the hypothesis that KL-control with a strong, pre-trained prior can be used to improve batch RL.

As shown in Figure 4.10, without KL-regularization the baseline RL models diverge quickly and continuously from the prior, losing information about realistic sequences. This figure also helps explain the poor performance of DBCQ in Table 4.5. The underlying Q-network

[2] We also compare the RL models to the prior, and see performance improvements in terms of the elicited human reward, but not in terms of the quality ratings. We believe this is because the rewards proposed here do not fully cover what it means to have a high quality conversation (as we will elucidate later), and hope that other researchers be able to use the techniques we propose to learn from improved rewards.

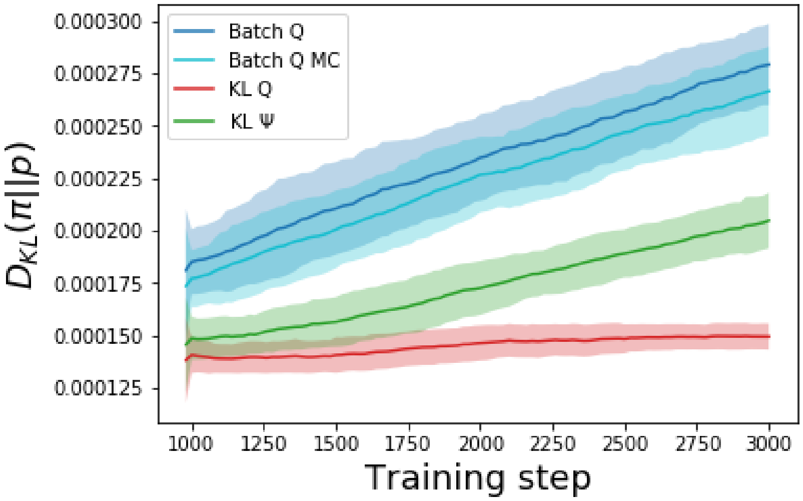


Figure 4.10: KL-divergence of the policy from the prior is lower with KL-control throughout training. Bands show standard deviation.

[User]: hey, what's up? **[Batch Q]**: where did you get??, that sounds so exciting. how do you work for you... you want to talk to me?... you want **[User]**: yes, i want to talk to you! **[Batch Q]**: where did you say to me?... you want to talk to you about you... you are so good for you... you are so good... **[User]**: thanks, i appreciate that. **[Batch Q]**: where did you get??, that sounds so exciting. how do you work for you... you want to see you...	**[User]**: hey, what's up? **[KL-control]**: i'm not sure, but i'm glad to see you, and i'm so happy that you're not alone **[User]**: oh thanks, that's very nice of you! **[KL-control]**: i'm sorry, but i'm glad to see what you think. **[User]**: don't be sorry, you did a good job. **[KL-control]**: that's really nice to me.

Table 4.6: Purely reward-maximizing methods like *Batch Q* trivially exploit the reward function by asking a question every turn, and using the maximum number of tokens in every sentence. Such models diverge away from realistic language, using implausible phrases. In contrast, KL-control methods output plausible language by staying close to the prior, but shift to using polite, cheerful language to maximize implicit human rewards.

in DBCQ does not directly integrate the prior. As Q-learning causes the model to diverge from the prior, the Q-estimates of language generated according to the prior become unrealistic, and Eq. 4.9 selects unrealistic actions. This results in highly 'diverse' (random) generated utterances. Although DBCQ performed well in simple domains in Section 4.2.5, it does not scale effectively to dialog in our experiments. Note that since we operate in discrete action space, we could not include the perturbation model originally proposed by (Fujimoto et al., 2018b), which may be critical to achieving good performance with BCQ.

The pre-trained prior may be especially important in a generative domain like dialog, where the true reward function is unknown, and so purely maximizing a heuristic may actually lead to lower quality conversations. Table 4.6 shows examples of conversations with a Batch Q and KL-control model. Because the Batch Q model has no incentive to stay close to realistic language, it learns to exploit the reward by asking a question and outputting the maximum number of tokens (30) every utterance. These sentences contain implausible phrases that do not represent realistic language (e.g. *"where did you say to me?*"). In contrast, the KL-control model uses fluent language, but shifts its distribution towards cheerful and polite speech, presumably because this is what led to positive human responses in the batch data. Rather than simply cherry-picking results, we invite the reader to check for themselves; all of the models tested in this study are available at: `https://neural.chat/rl`.

In fact, we noticed that all models trained with the implicit human rewards described in Section 4.2.6.1 learned to use more cheerful and supportive language. Therefore, we create post-hoc metrics to measure this effect (see the Appendix for details). Figure 4.11 shows how these metrics, as well as the implicit rewards, differ across models. Without KL-control, baseline methods like Batch Q exploit simple rewards like asking questions at the expense of realistic language, explaining their poor quality ratings. In contrast, KL-control models learn to rely more on realistic but polite, supportive, and cheerful dialog to elicit higher total *human reward*.

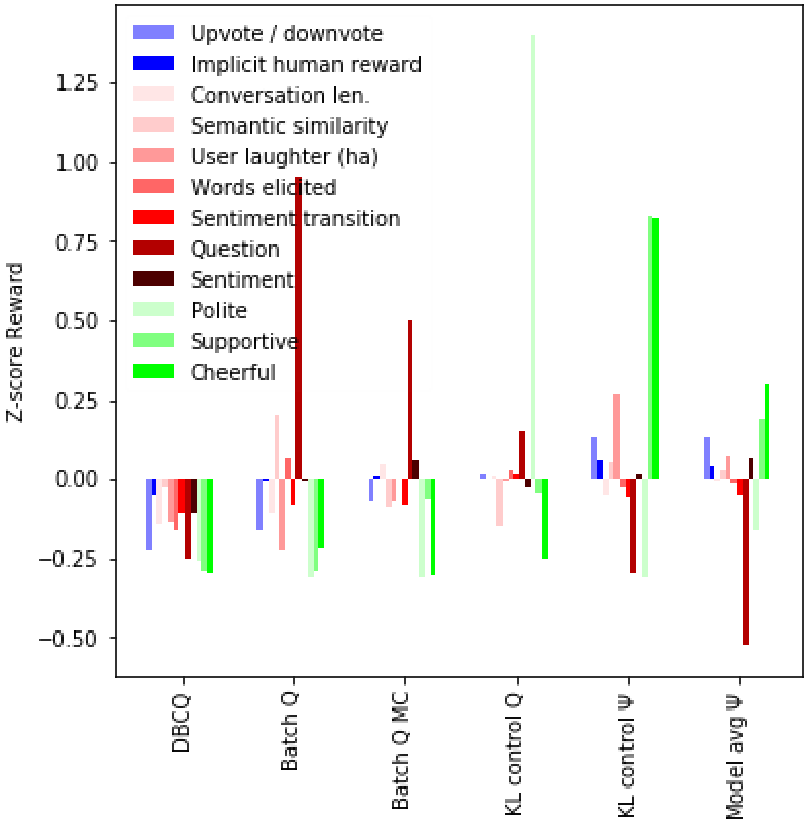


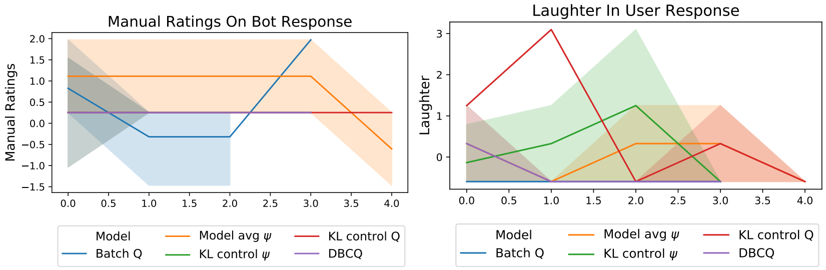
Figure 4.11: Z-scored reward. Red metrics were used in training rewards, green are post-hoc. Traditional RL methods like Batch Q exploit simple action-based rewards, like asking questions. In contrast, KL-control methods shift their distribution towards polite, supportive, and cheerful conversation, allowing them to elicit higher *human reward* (blue).

Reward function	Quality	Fluent	Diverse	Related	Empathy	Total	Votes	Human reward
Conv. len.	2.20 ±.40	3.61 ±.53	3.02 ±.52	2.25 ±.46	2.48 ±.45	13.57 ±1.84	-.035	-.003
Semantic sim.	1.93 ±.34	3.50 ±.45	2.37 ±.45	2.11 ±.45	2.52 ±.48	12.43 ±1.75	-.020	.012
User laughter	1.96 ±.38	3.56 ±.48	2.33 ±.51	1.93 ±.42	3.20 ±.55	12.98 ±1.60	-.149	-.003
Words elicited	2.11 ±.32	3.96 ±.44	3.04 ±.45	2.04 ±.35	2.55 ±.46	13.70 ±1.44	.059	.024
Manual votes	2.14 ±.38	3.47 ±.45	2.91 ±.47	2.07 ±.39	2.42 ±.46	13.00 ±1.65	-.030	.010
Sent. trans.	2.02 ±.31	3.71 ±.49	2.98 ±.50	2.04 ±.42	2.84 ±.48	13.60 ±1.63	.031	.014
Question	2.29 ±.37	**4.31 ±.50**	**3.31 ±.52**	2.20 ±.40	2.60 ±.41	14.71 ±1.63	.057	.012
Sentiment	**2.47 ±.32**	4.05 ±.45	3.23 ±.46	**2.42 ±.39**	**3.23 ±.55**	**15.40 ±1.49**	**.085**	**.045**

Table 4.7: Interactive human evaluation of WOP trained with different reward functions. Sentiment leads to the highest quality. Learning from manual button presses is not as effective as learning from implicit metrics like sentiment.

Table 4.7 presents the results of WOP trained with only a single reward function, ordered from lowest to highest quality. Notably, extracting multiple different reward functions post-hoc from a batch of data and training on these independently is only possible with an effective BRL model. Here all models are trained with KL-control, Ψ-learning, and MC targets. Investigating which rewards presented in Section 4.2.6.1 are most critical to achieving high-quality conversations with humans, we note that maximizing positive and minimizing negative sentiment in the user turns out to lead to the highest quality bot. This underscores the importance of affective signals as cues for good conversation. Bots trained on the manual upvotes and downvotes provided by users on the utterance level fail to achieve similarly high performance. Even though users were instructed to make use of the vote feature, the task is burdensome, and users did not vote frequently enough to provide a good training signal. This validates the hypothesis that *implicit* signals of human enjoyment (such as sentiment) are a more scalable way to learn from human preferences.

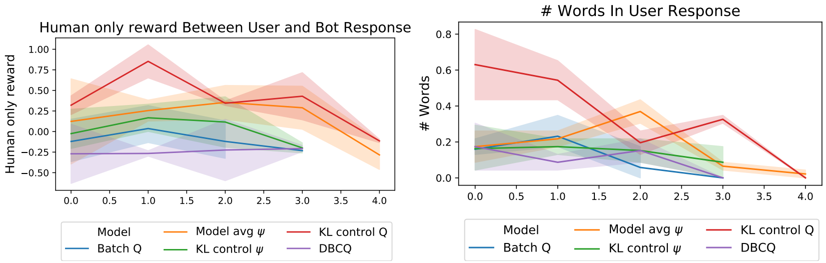
To further elucidate the effect of the implicit rewards, Figure 4.12 shows the reward trajectory over the ten best conversations obtained with models trained with different techniques. Figure 4.12 (a) shows that manual votes are indeed a rare event, with only the best and worst models receiving an upvote or downvote. Once again, this suggests that explicit feedback from humans is a cumbersome and sparse reward signal. As shown in 4.12 (b), eliciting laughter is an extremely rare event, and only the KL-control models are able to do so. While we see that KL-control models are able to elicit significantly higher reward than baselines, we note that KL-control Q performs best overall and in terms of words elicited, even though it had lower quality ratings in Table 4.5. This suggests that maximizing these rewards is not a perfect proxy for human judgments of quality. Note that this aligns with what we observed when we compared the RL models


Manual Ratings On Bot Response

Laughter In User Response

(a) Manual votes

(b) Laughter


Human only reward Between User and Bot Response

Words In User Response

(c) Implicit human reward

(d) Words elicited

Figure 4.12: Comparison of top 10 conversation trajectories observed across deployed models, showing the 90% confidence interval of the rewards.

in Table 4.5 to the pre-trained prior. While we saw performance improvements in terms of the elicited human reward, the human quality ratings were not significantly better. We believe this is because the rewards proposed here do not fully cover what it means to have a high quality conversation. We hope that other researchers will be able to use the techniques we propose here to learn effectively from improved rewards in an off-policy batch setting.

4.2.8 Conclusion

This section presents the Way Off-Policy (WOP) algorithm, which improves performance when learning off-policy without the possibility to explore – i.e. batch RL (BRL). We are the first to propose using KL-control from a strong prior model pre-trained on data as a way to avoid extrapolation error and instability in BRL. Our results on traditional RL tasks demonstrate that WOP provides performance improvements over state-of-the-art BRL techniques, and results in dialog generation show that KL-control is critical to achieving good performance in this real-world, high-dimensional setting. In a generative domain such as dialog, the true reward function is not known, and trivially exploiting the rewards can actually lead to worse performance. Thus, KL-control may be particularly necessary to ensure samples remain realistic and close to the data distribution. We propose several reward functions that could allow an open-domain dialog generation model to learn from rich cues implicit in human interaction, where learning from expressed sentiment was most promising. While these rewards are far from perfect or complete, we see that maximizing implicit rewards leads to better performance than relying on explicit feedback. We hope that the techniques presented here can improve learning with RL from offline data, making it easier to apply RL to safety-critical settings such as human interaction.

4.3 KL-control for improved sequence generation with RL

The KL-control approach proposed in the previous section—specifically, minimizing KL-divergence from a strong prior pre-trained on sequence data—is broadly applicable, and can be useful in a variety of domains. Below, we briefly present the results of additional experiments using the KL-regularized Q-learning and Ψ-learning models presented above in the domains of music generation and drug discovery. Once again, we penalize KL-divergence from a pre-trained prior sequence model of $p(a|s)$, but here trained on sequences of musical notes or sequential encodings of drug molecules into text strings. We also introduce an additional method, called Recurrent G-learning, which directly incorporates the prior into the policy when choosing actions. These results emphasize the flexibility of the proposed approach, and its promise for learning from sparse human feedback data in a variety of domains.

4.3.1 Recurrent G-learning

We can derive another algorithm by parametrizing Ψ_θ indirectly by $\Psi_\theta(s_t, a_t) = \log p(a_t|s_t) + G_\theta(s_t, a_t)$. Substituting into the equations in Section 4.2.4.4, we get a different temporal-difference method:

$$\begin{aligned} &L_G(\theta) = \mathbb{E}_\beta[(G_\theta(s_t, a_t) - y_t)^2] \text{ where} \\ &y_t = r(s_t, a_t)/c + \gamma \log \sum_{a'} p(a'|s_{t+1}) e^{G(s_{t+1}, a')} \text{ and} \\ &\pi_\theta(a_t|s_t) \propto p(a_t|s_t) \exp(G_\theta(s_t, a_t)) \end{aligned} \tag{4.15}$$

This formulation corresponds to G-learning (Fox et al., 2016), which can thus be seen as a special case of generalized Ψ-learning. Unlike Ψ learning, which directly builds knowledge about the prior policy into the Ψ function, the G-function does not give the policy directly but instead needs to be dynamically mixed with the prior policy probabilities. While this computation is straight-forward for discrete action domains as here, extensions to continuous action domains require additional considerations such as normalizability of Ψ-function parametrization (Gu et al., 2016).

The KL control-based derivation also has another benefit in that the stochastic policies can be directly used as an exploration strategy, instead of heuristics such as ϵ-greedy or additive noise (Mnih et al., 2013; Lillicrap et al., 2015).

4.3.2 *KL-control for sequence generation*

Thus, we now have three methods for generating sequences which combine training on data and fine-tuning with RL using KL-control: Q-learning with log prior augmentation (based on Eq. 4.11), generalized Ψ-learning (based on Eq. 4.13), and G-learning (based on Eq. 4.15). These methods enjoy the benefits of both data and RL training, and are ideal in scenarios when neither approach will lead to ideal performance. This is often the case in sequence generation, since Maximimum Likelihood Estimation (MLE) training can lead to boring, repetitive, and unstructured output Li et al. (2016c), and it may be difficult to write reward functions that fully describe good task performance (as we saw in the previous section).

In the following experiments, we compare these approaches to both the original performance of the MLE RNN, and a model trained using only RL and no prior policy. Model evaluation is performed every 100,000 training epochs, by generating 100 sequences and assessing the average task reward r_T and prior probability $\log p(a|s)$[3].

4.3.3 *Experiment I: Melody generation*

Music compositions adhere to relatively well-defined structural rules, making music an interesting sequence generation challenge. For example, music theory tells that groups of notes belong to keys, chords follow progressions, and songs have consistent structures made up of musical phrases. Our research question is therefore whether such constraints can be learned by an RNN, while still allowing it to maintain note probabilities learned from data.

[3] The code is available in open-source at `https://github.com/tensorflow/magenta/tree/master/magenta/models/rl_tuner`.

To test this hypothesis, we developed several rules that we believe describe pleasant-sounding melodies, taking inspiration from a text on melodic composition (Gauldin, 1995). We do not claim these characteristics are exhaustive or strictly necessary for good composition; rather, they are an incomplete measure of task success that can simply guide the model towards traditional composition structure. It is therefore crucial that the KL-control approach allows the model to retain knowledge learned from real songs in the training data. The rules comprising the music-specific reward function $r_T(a, s)$ encourage melodies to: stay in key, start with the tonic note, resolve melodic leaps, have a unique maximum and minimum note, prefer harmonious intervals, play motifs and repeat them, have a low autocorrelation at a lag of 1, 2, and 3 beats, and avoid excessively repeating notes. Interestingly, while excessively repeating tokens is a common problem in RNN sequence generation models, avoiding this behavior is also Gauldin's first rule of melodic composition (p. 42).

To train the model, we begin by extracting monophonic melodies from a corpus of 30,000 MIDI songs and encoding them as one-hot sequences of notes[4]. These melodies are then used to train an LSTM with one layer of 100 cells. Optimization was performed with Adam (Kingma and Ba, 2014), a batch size of 128, initial learning rate of 0.5, and a stepwise learning rate decay of 0.85 every 1000 steps. Gradients were clipped to ensure the L2 norm was less than 5, and weight regularization was applied with $\beta = 2.5 \times 10^{-5}$. Finally, the losses for the first 8 notes of each sequence were not used to train the model, since it cannot reasonably be expected to accurately predict them with no context. The trained RNN eventually obtained a validation accuracy of 92% and a log perplexity score of 0.2536. This model was used as described above to initialize the three sub-networks in the KL-control models.

The KL-control models were trained using a similar configuration to the one above, except with a batch size of 32, and a reward discount factor of γ=0.5. The Target-Q-network's weights θ^- were gradually updated towards those of the Q-network (θ) according to the formula $(1 - \eta)\theta^- + \eta\theta$, where $\eta = 0.01$ is the Target-Q-network update rate. A strength of our model is that the influence of data and task-specific rewards can be explicitly controlled by adjusting the weighting parameter c. We replicated our results for a number of settings for c; we present results for $c = .5$ below because we believe them to be most musically pleasing, however additional results are available at `https://goo.gl/cTZy8r`. Similarly, we replicated the results using both ϵ-greedy and Boltzmann exploration, and present the results

[4] More information about both the note encoding and the reward metrics is available in Jaques et al. (2017a)

using ϵ-greedy exploration below.

4.3.3.1 *Melody generation results*

Table 4.8 provides quantitative results in the form of performance on the music theory rules to which we trained the model to adhere; for example, we can assess the fraction of notes played by the model which belonged to the correct key, or the fraction of melodic leaps that were resolved. The statistics were computed by randomly generating 100,000 melodies from each model.

Metric	**MLE**	**Q**	**Ψ**	**G**
Repeated notes	63.3%	**0.0%**	**0.02%**	**0.03%**
Mean autocorr. lag 1	-.16	**-.11**	**-.10**	.55
Mean autocorr. lag 2	.14	**.03**	**-.01**	.31
Mean autocorr. lag 3	-.13	**.03**	**.01**	17
Notes not in key	0.1%	1.00%	0.60%	28.7%
Starts with tonic	0.9%	**28.8%**	**28.7%**	0.0%
Leaps resolved	77.2%	**91.1%**	**90.0%**	52.2%
Unique max note	64.7%	56.4%	59.4%	37.1%
Unique min note	49.4%	51.9%	**58.3%**	**56.5%**
Notes in motif	5.9%	**75.7%**	**73.8%**	**69.3%**
Notes in repeat motif	0.007%	**0.11%**	**0.09%**	0.01%

Table 4.8: Statistics of music theory rule adherence based on 100,000 randomly initialized melodies generated by each model. The top half of the table contains metrics that should be minimized, while the bottom half contains metrics that should increase. Bolded entries represent significant improvements over the MLE baseline.

The results above demonstrate that the application of RL is able to correct almost all of the targeted "bad behaviors" of the MLE RNN, while improving performance on the desired metrics. For example, the original LSTM model was extremely prone to repeating the same note; after applying RL, we see that the number of notes belonging to some excessively repeated segment has dropped from 63% to nearly 0% in all of the KL-control models. While the metrics for the G model did not improve as consistently, the Q and Ψ models successfully learned to adhere to most of the imposed rules. The degree of improvement on these metrics is related to the magnitude of the reward given for the behavior. For example, a strong penalty of -100 was applied each time a note was excessively repeated, while a reward of only 3 was applied at the end of a melody for unique extrema notes (which most likely explains the lack of improvement on this metric). The reward values could be adjusted to improve the metrics further, however we found that these values produced pleasant melodies.

While the metrics indicate that the targeted behaviors of the RNN have improved, it is not clear whether the models have retained information about the training data. Figure 4.13 plots the average $\log p(a|s)$ as produced by the Reward RNN for melodies generated by the models every 100,000 training epochs. Included in the plot is an *RL only* model trained using only the music theory rewards, with no information about $\log p(a|s)$. Since each model is initialized with the weights of the trained MLE RNN, we see that as the models quickly learn to adhere to the music theory constraints, $\log p(a|s)$ falls from its initial point. For the RL only model, $\log p(a|s)$ reaches an average of -3.65, which is equivalent to an average $p(a|s)$ of approximately 0.026, or essentially a random policy over the 38 actions with respect to the distribution defined by the Reward RNN. This is strong evidence of catastrophic forgetting; even though the *RL-only* model was initialized with a prior that had effectively learned from data, this information is lost when naively continuing to train with RL. However, as shown in Figure 4.13, the KL-control models (Q, Ψ, and G) attain higher $\log p(a|s)$ values than this baseline even over 3,000,000 training steps. This indicates they have maintained information about the data distribution throughout training. The G-learning implementation scores highest on this metric, at the cost of slightly lower average r_T.

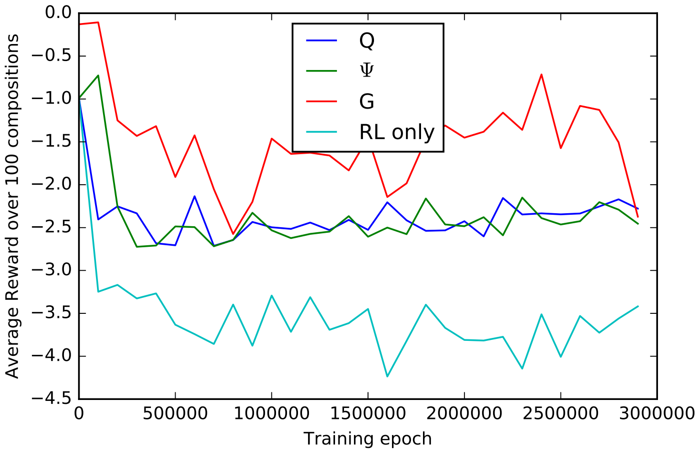


Figure 4.13: Average prior reward $p(a|s)$ (probability of notes under the MLE prior), obtained by sampling 100 melodies every 100,000 training epochs. The three models are compared to a model trained using only the music theory rewards r_T. This *RL-only* model shows evidence of catastrophic forgetting, diverging to a policy that is random with respect to the prior. In contrast, KL-control alleviates catastrophic forgetting.

This compromise between data probability and adherence to music theory could explain the difference in the G model's performance on the music theory metrics in Table 4.8. Figure 4.14 plots the average r_T, which also shows that the G model obtains less task reward. Finally, we have verified that by increasing the c parameter it is possible to train all the models to have even higher average $\log p(a|s)$, but found that $c = 0.5$ produced melodies that sounded better subjectively.

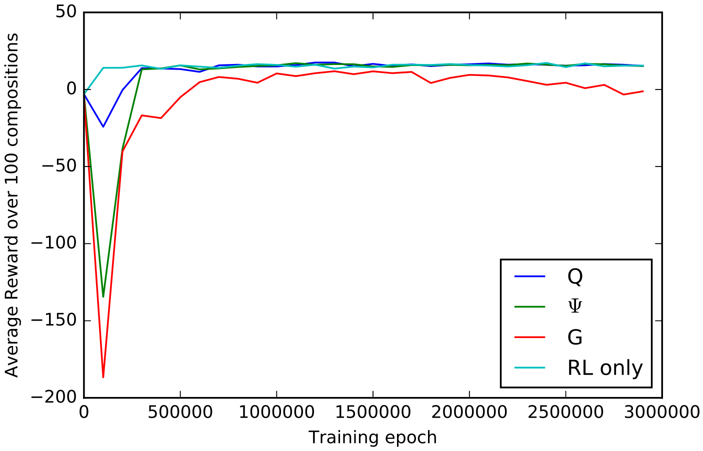


Figure 4.14: Average music theory reward obtained by sampling 100 melodies every 100,000 training epochs. The three models are once again compared to a model trained using only the music theory rewards r_T.

The question remains whether the RL-tutored models actually produce more pleasing melodies. The sample melodies used for the study are available here: `goo.gl/XIYt9m`; we encourage readers to judge their quality for themselves. To more formally answer this question, we conducted a user study via Amazon Mechanical Turk in which participants were asked to rate which of two randomly selected melodies they preferred on a Likert scale. A total of 192 ratings were collected; each model was involved in 92 of these comparisons. Figure 4.15 plots the number of comparisons in which a melody from each model was selected as the most musically pleasing. A Kruskal-Wallis H test of the ratings showed that there was a statistically significant difference between the models, $\chi^2(3) = 109.480, p < 0.001$. Mann-Whitney U post-hoc tests revealed that the melodies from all three Sequence Tuner models (Q, Ψ, and G) had significantly higher ratings than the melodies of the MLE RNN, $p < .001$. The Q and Ψ melodies were also rated as significantly more pleasing than those of the G model, but did not differ significantly from each other.

4.3.3.2 *Discussion*

Listening to the samples produced by the MLE RNN reveals that they are sometimes dischordant and usually dull; the model tends to place rests frequently, repeat the same note, and produce melodies with little variation. In contrast, the melodies produced by the KL-control models are more varied and interesting. The G model tends to produce energetic and chaotic melodies, which include sequences of repeated notes. This repetition is likely because the G policy as

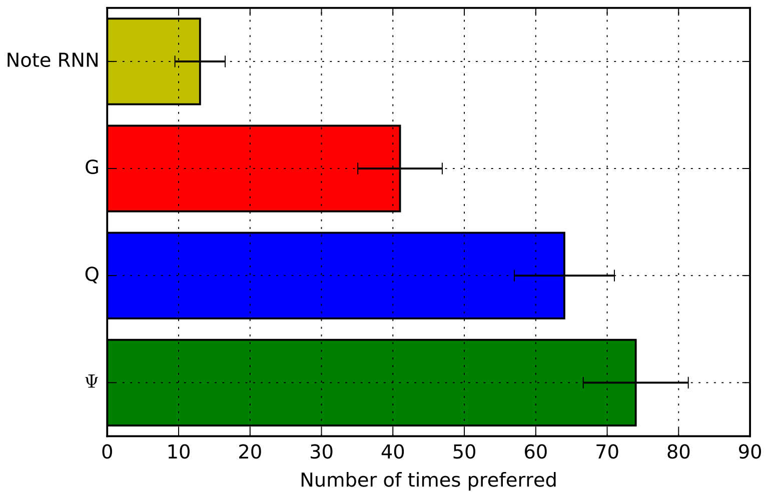


Figure 4.15: The number of times a melody from each model was selected as most musically pleasing. Error bars reflect the std. dev. of a binomial distribution fit to the binary win/loss data from each model. KL-control models G, Q, and Ψ are significantly preferred over the prior.

defined in Eq. 4.15 directly mixes $p(a|s)$ with the output of the G network, and the MLE RNN strongly favours repeating notes. The most pleasant melodies are generated by the Q and Ψ models. These melodies stay firmly in key and frequently choose more harmonious interval steps, leading to melodic and pleasant tunes. However, it is clear they have retained information about the training data; for example, the sample q2.wav in the sample directory ends with a seemingly familiar riff.

While we acknowledge that the monophonic melodies generated by these models—which are based on highly simplistic rules of melodic composition—do not approach the level of artistic merit of human composers, we believe this study provides a proof-of-concept that encoding even incomplete and partially specified domain knowledge using our method can help the outputs of an LSTM adhere to a more consistent structure. The musical complexity of the songs is limited not just by the heuristic rules, but also by the simple monophonic encoding, which cannot represent the dynamics and expressivity of a musical performance. Although these melodies cannot surpass those of human musicians, they provide a way to train a model to generate aesthetically pleasing outputs in the absence of a better metric of human taste than log-likelihood. Ultimately, we would like to use human feedback as the ideal reward signal, dynamically learning from human reactions to improve the model's ability to generate music. For example, it might be possible to use Electrodermal Activity (EDA) to sense when a person finds music especially stimulating (or gets goosebumps!), and use these techniques to train the model to generate songs that are more likely to elicit such reactions.

4.3.4 *Experiment II: Computational drug discovery*

As a further experiment, we tested the effectiveness of Sequence Tutor for generating a higher yield of synthetically accessible drug-like molecules. Organic molecules can be encoded using the commonly used SMILES representation (Weininger, 1970). For example, amphetamine can be encoded as 'CC(N)Cc1ccccc1', while creatine is 'CN(CC(=O)O)C(=N)N'. Using this character encoding, it is straightforward to train an MLE RNN to generate sequences of SMILES characters (e.g. Segler et al. (2017)); we trained such a model using the same settings as described above for the melody MLE RNN. However, only about a third of the molecules generated using this simple approach are actually valid SMILES encodings. Further, this approach does not directly optimize for metrics of molecule or drug quality. These metrics include: a) the water-octanol partition coefficient (logP), which is important in assessing the drug-likeness of a molecule; b) synthetic accessibility (SA) (Ertl and Schuffenhauer, 2009), a score from 1-10 that is lower if the molecule is easier to synthesize; and c) Quantitative Estimation of Drug-likeness (QED) (Bickerton et al., 2012), a more subjective measure of drug-likeness based on abstract ideas of medicinal aesthetics.

To optimize for these metrics, while simultaneously improving the percent yield of valid molecules from the RNN, we constructed a reward function that incentivizes validity, logP, SA, and QED using an open-source library called RDkit[5]. Included in the reward function was a penalty for molecules with unrealistically large carbon rings (size larger than 6), as per previous work (Gómez-Bombarelli et al., 2016). Finally, after observing that the model could exploit the reward function by generating the simple molecule 'N' repeatedly, or 'CCCCC...' (which produces an unrealistically high logP value), we added penalties for sequences shorter than, or with more consecutive carbon atoms than, any sequence in the training data. The KL-control models were then trained using these rewards, the pre-trained MLE RNN, and similar settings to the first experiment, except with ϵ-greedy exploration with $\epsilon = .01$, a batch size of 512, and discount factor $\gamma = .95$. For this experiment, we also made use of prioritized experience replay (Schaul et al., 2015) to allow the model to more frequently learn from relatively rare valid samples. A value of $c =$

[5] `http://www.rdkit.org/`

2.85 led to a higher yield of valid molecules with high metrics, but still encouraged the diversity of generated samples.

4.3.4.1 *Drug discovery results and discussion*

As the Ψ algorithm produced the best results for the music generation task, we focused on using this technique for generating molecules. Table 4.9 shows the performance of this model against the original MLE model according to metrics of validity, drug-likeness, and synthetic accessibility. Once again, KL-control is able to significantly improve almost all of the targeted metrics. However, it should be noted that the KL-control models tend to produce simplistic molecules involving more carbon atoms than the MLE baseline; e.g. Sequence Tutor may produce 'SNCc1ccccc1', while the MLE produces 'C(=O)c1ccc(S(=O)(=O)N(C)C)c(Cl)c1', which is the reason for the Sequence Tutor model's lower QED scores. This effect is due to the fact that simple sequences are more likely to be valid, have high logP and SA scores, and carbon is highly likely under the distribution learned by the MLE model. A higher reward for QED and further improvement of the task-specific rewards based on domain knowledge could help to alleviate these problems. Overall, the fact that KL-control can improve the percentage of valid molecules produced as well as the logP and synthetic accessibility scores serves as a proof-of-concept that Sequence Tutor may be valuable in a number of domains for imparting domain knowledge onto a sequence predictor.

Metric	**MLE**	**Q**
Percent valid	30.3%	**35.8%**
Mean logP	2.07	**4.21**
Mean QED	.678	.417
Mean SA penalty	-2.77	**-1.79**
Mean ring penalty	-.096	**-.001**

Table 4.9: Statistics of molecule validity and quality based on 100,000 randomly initialized samples. Bolded entries represent significant improvements over the MLE baseline.

4.3.5 *Conclusion and future work*

We have derived a novel sequence learning framework which uses RL to correct properties of sequences generated by an RNN, while maintaining information learned from MLE training on data, and ensuring the diversity of generated samples. By demonstrating a connection between our sequence generation approach and KL-control, we have derived three novel RL-based methods for optimizing sequence generation models. These methods were empirically compared in the context of a music generation task, and further demonstrated on a computational molecular generation task. KL-control from a pre-trained prior showed promising results in terms of both adherence to task-specific rules, and subjective quality of the generated sequences.

We believe the approach of using RL to refine RNN models could be promising for a number of applications, including the reduction of bias in deep learning models. While manually writing a domain-specific reward function may seem unappealing, that approach is limited by the quality of the data that can be collected, and besides, even state-of-the-art sequence models often fail to learn all the aspects of high-level structure (van den Oord et al., 2016; Graves, 2013). Further, the data may contain hidden biases, as has been demonstrated for popular language models (Caliskan-Islam et al., 2016). In contrast to relying solely on possibly biased data, our approach allows for encoding high-level domain knowledge into the RNN, providing a general, alternative tool for training sequence models. Finally, the most promising use for these techniques is to learn from human social responses in order to fine-tune models with scarce and limited samples. This could potentially allow a wide range of machine learning models to incorporate human feedback.

4.4 *Statement of contributions*

Sections 4.1 and 4.2 of this chapter are the results of a large, collaborative effort on dialog systems research in Rosalind Picard's Affective Computing group. I initially began the project myself in October 2018 by familiarizing myself with the relevant literature, and began asking members of the group if they would like to work on it with me. By February 2019, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, and Noah Jones had agreed to participate. I am grateful to all of these co-authors for a truly collaborative project involving deep sharing of ideas and joint effort. The first section describes the results of training and evaluating many dialog models using an interactive online platform. My role in this first project was to formulate the initial ideas, including incorporating sentiment (DeepMoji), using variational hierarchical models, and the idea for EI regularization, which were then extended and improved by all authors. Asma proposed the idea of evaluating with self-play. I wrote the code that allowed interacting with the models and training with EI regularization, and handled the majority of the effort related to training the models and deploying them to the web server. Craig wrote the code for the web server itself, and designed the user interface. Asma wrote code to integrate DeepMoji and Infersent, after investigating other approaches to topic modeling. Judy handled running the interactive study on Mechanical Turk. Noah advised on the psychological aspects of empathy and good conversation. Agata and Rosalind provided advice and guidance throughout.

Learning from implicit human reactions in conversation with RL (Section 4.2) was an idea I originally proposed during my thesis proposal, and this part of the project was something that I worked on independently, although still consulting with Asma, Judy, Rosalind, and Shane Gu. I formulated the idea of improving Batch RL with KL-control from a pre-trained prior after considering how to work with a static dataset and reading Scott Fujimoto's BCQ paper. Scott also provided advice on the project over email. The code and experiments for this section were written and conducted by me. Craig and Judy helped run the evaluation of the RL models on the server and on Mechanical Turk.

Section 4.3 resulted from an earlier paper, which began as part of an internship project supervised by Douglas Eck. I initially wrote code and ran experiments to improve a music generation model with Q-learning by adding the prior probability $p(a|s)$ to the reward function. The music-theory rewards were developed with advice from Curtis 'Fjord' Hawthorne. After presenting the internship project, Shane Gu approached me about developing some theoretical improvements to my technique, and was instrumental in developing the Ψ-learning and G-learning formulations of the algorithm. Shane had the idea to extend the experiments to the drug discovery domain and put me in touch with José Hernández-Lobato, who provided advice as I wrote the code and ran the experiments for drug discovery. Dzmitry Bahdanau provided initial consultation about Q-learning and the original algorithm, and advice and feedback on the manuscript.

4.5 Appendix

4.5.1 Self-play hybrid metric coefficients

We optimized the coefficients of sub-components of the hybrid metric using a leave-bot-out scenario. As shown in Figure 4.16, we observe that λ_is are stable across these training iterations. However, because

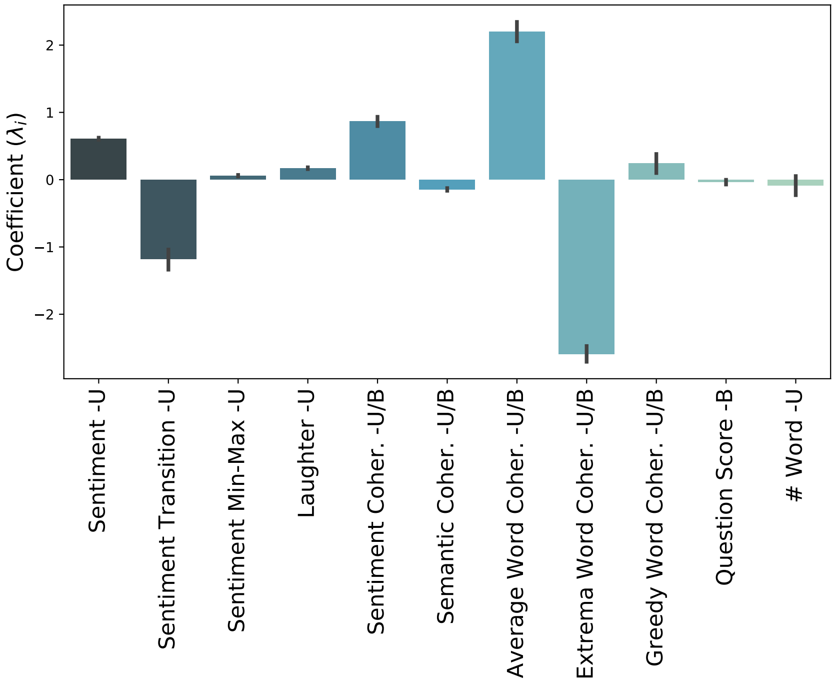


Figure 4.16: The learned coefficients (λ_i) that the hybrid metric (M_H) is comprised of. Using a leave-bot-out method, we observe that the λ_is are stable. The error bars show 90% confidence intervals.

we have optimized a linear regression equation and some of the features have overlapping information, such as different aggregation methods for calculating word coherence, we do not suggest using λ_is for direct interpretation; further investigation is required.

4.5.2 *Reddit casual conversation corpus details*

Using the 1.7 Billion post comments dataset hosted on Google BigQuery, we extracted post ids for all posts on `https://www.reddit.com/r/CasualConversation` from July 2018 to December 2018. For each post, we built a conversation tree of comments and subsequent replies to extract three-turn dialog. We removed links, excluded *[removed]* and *[deleted]* tag comments, and only used text before "*edit*" comments to preserve the original content in the conversation. We make this dataset available for public use at `https://affect.media.mit.edu/neural_chat/datasets`.

4.5.3 *Embedding-based metrics*

Embedding Average Taking the mean word embedding of the generated sentence e_g and the target sentence e_t, the embedding average metric is the cosine distance between the two.

$$\bar{e}_t = \frac{\sum_{w \in t} e_w}{|\sum_{w' \in t} e_{w'}|} \tag{4.16}$$

$$\textsc{Avg}(\hat{e}_t, \hat{e}_g) = cos(\bar{e}_t, \bar{e}_g) \tag{4.17}$$

Vector Extrema The extrema vector for a sentence can be calculated by taking the most extreme value for each dimension ($e_w^{(d)}$) among the word vectors in the sentence. The extrema embedding metric is again the cosine distance between the extrema sentence vectors.

$$\hat{e}_t^{(d)} = \begin{cases} \max_{w \in t} e_w^{(d)} & \text{if } e^{(d)} > |\min_{w' \in t} e_{w'}^{(d)}| \\ \min_{w \in t} e_w^{(d)} & \text{otherwise} \end{cases} \tag{4.18}$$

$$\textsc{Ext}(\hat{e}_t, \hat{e}_g) = cos(\hat{e}_t, \hat{e}_g) \tag{4.19}$$

Greedy Matching The greedy matching distance is computed by matching word vectors in a source sentence (s) with the closest words vectors in the target sentence(s).

$$G(r, \hat{r}) = \frac{\sum_{w \in r;} \max_{\hat{w} \in \hat{r}} cos(e_w, e_{\hat{w}})}{|r|} \tag{4.20}$$

$$\textsc{Grd}(s, t) = \frac{G(s,t) + G(t,s)}{2} \tag{4.21}$$

4.5.4 *RL post-hoc metrics*

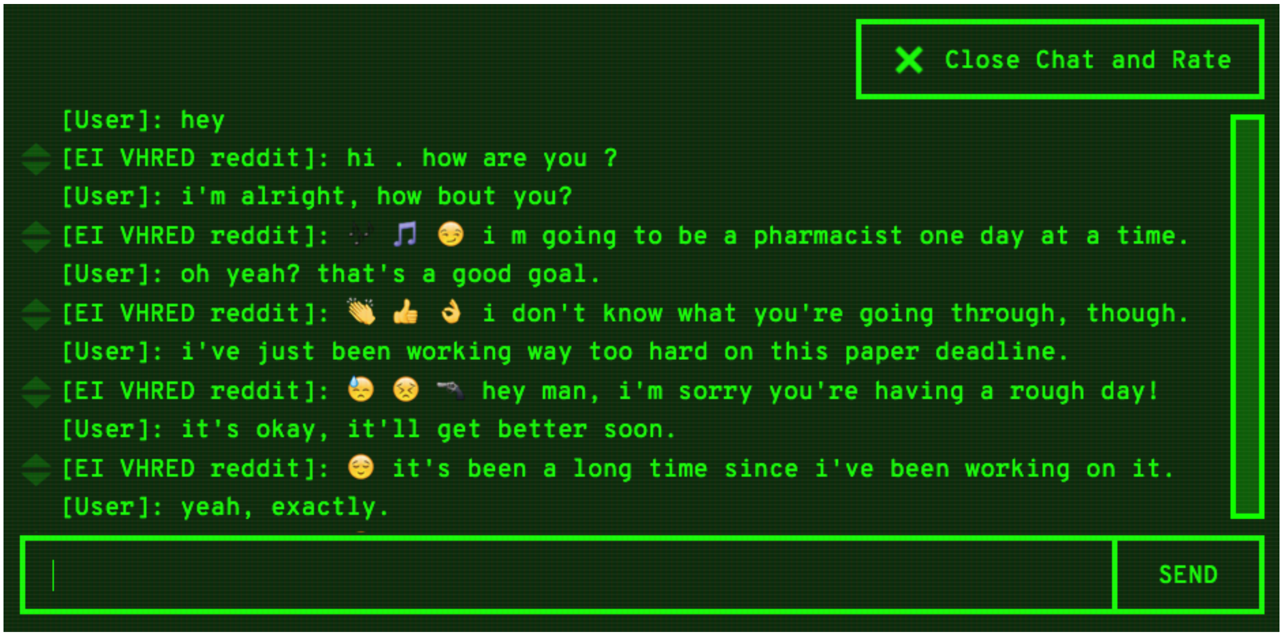
After training the bots on the above rewards, we noticed a shift in the distribution of their language towards more polite, cheerful, and supportive speech. Therefore, we designed post-hoc metrics to measure these qualities, which are based on counting whether a subset of phrases is present in an utterance.

Politeness phrases: *if I may; may I; please; thanks; no worries; if you don't mind; have a great day; I'm sorry.*

Supportive phrases: *you're right; you are right; you're not alone; you are not alone; congrats; that's a good idea; that is a good idea; you'll be fine; you will be fine; you'll be okay; you will be okay; it will get better; sorry you're* *going through; sorry you are going through; if it makes you feel better; if it makes you feel any better; keep your head up; keep it up; I'm in a similar situation; I am in a similar situation; you'll get it; you will get it; happy for you; I'm in the same boat; I am in the same boat; if you feel like you need to vent.*

Cheerful phrases: *nice to hear; happy; excited; really nice; glad; the best; great; good time; looking forward; beautiful.*

4.5.5 *Interactive evaluation details*

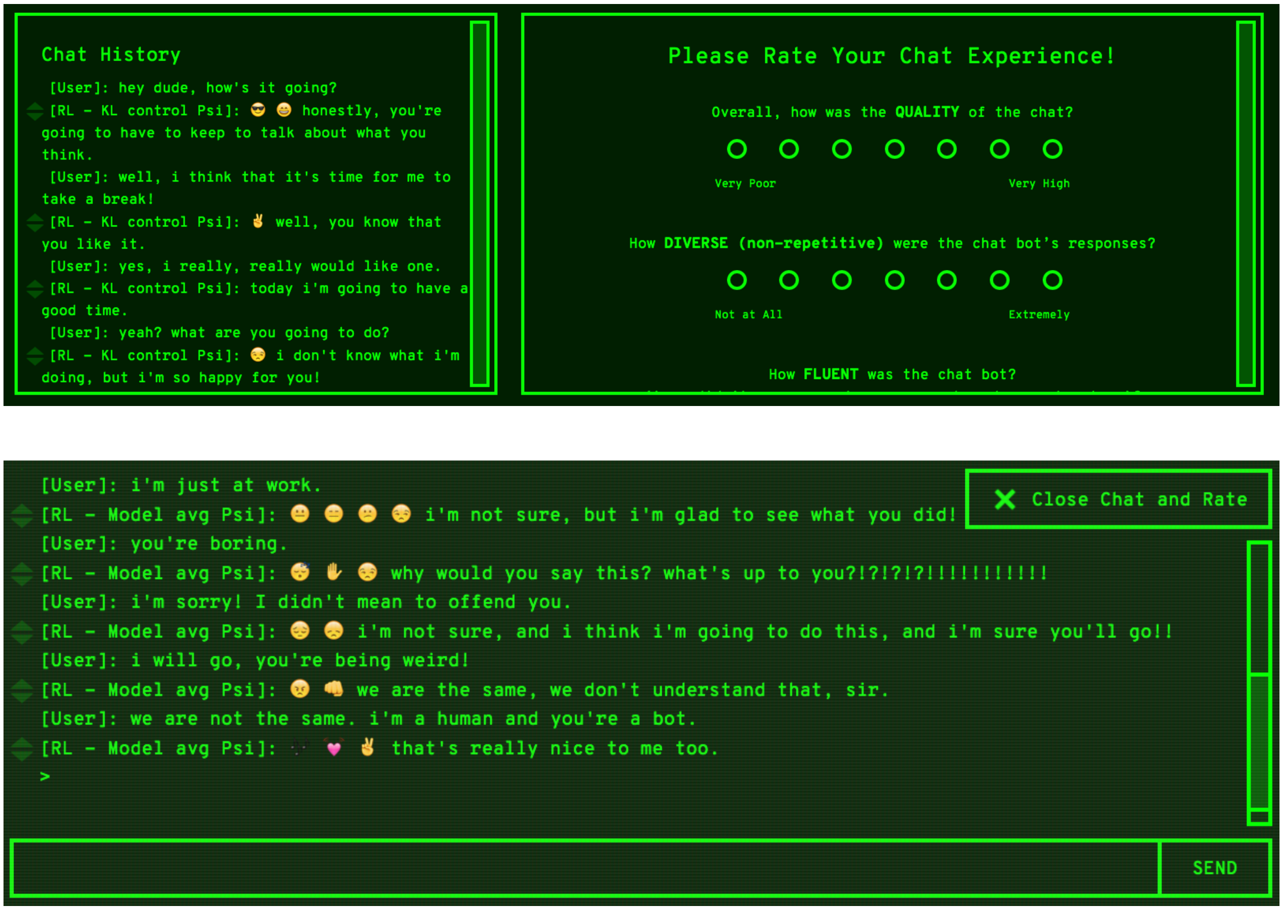


[User]: hey
[EI VHRED reddit]: hi . how are you ?
[User]: i'm alright, how bout you?
[EI VHRED reddit]: 🎶 🎵 😏 i m going to be a pharmacist one day at a time.
[User]: oh yeah? that's a good goal.
[EI VHRED reddit]: 👏 👍 👌 i don't know what you're going through, though.
[User]: i've just been working way too hard on this paper deadline.
[EI VHRED reddit]: 😓 😣 🔫 hey man, i'm sorry you're having a rough day!
[User]: it's okay, it'll get better soon.
[EI VHRED reddit]: 😌 it's been a long time since i've been working on it.
[User]: yeah, exactly.

Figure 4.17: Interactive evaluation chat interface

For our interactive evaluation, we built a platform to mimic a natural chat setting. Figure 4.17 is an example conversation within the platform that interactive evaluation participants see. Annotators can optionally click the up and down arrows beside each chatbot response to give feedback on the specific utterance. Once 3 or more turns of the conversation has taken place, participants may click "Close Chat and Rate". This will take them to the rating page where the conversations to be rated is presented along side the 7 point Likert scale questions used to asses the conversation (Figure 4.2). Figure4.18 shows example conversations with the RL models.

For the experiments presented in Section 4.1, participants both from Amazon Mechanical Turk and from the authors' institution were



Chat History

[User]: hey dude, how's it going?
[RL - KL control Psi]: 😎 😁 honestly, you're going to have to keep to talk about what you think.
[User]: well, i think that it's time for me to take a break!
[RL - KL control Psi]: ✌️ well, you know that you like it.
[User]: yes, i really, really would like one.
[RL - KL control Psi]: today i'm going to have a good time.
[User]: yeah? what are you going to do?
[RL - KL control Psi]: 😒 i don't know what i'm doing, but i'm so happy for you!

Please Rate Your Chat Experience!

Overall, how was the **QUALITY** of the chat?
○ ○ ○ ○ ○ ○ ○
Very Poor … Very High

How **DIVERSE (non-repetitive)** were the chat bot's responses?
○ ○ ○ ○ ○ ○ ○
Not at All … Extremely

How **FLUENT** was the chat bot?

[User]: i'm just at work.
[RL - Model avg Psi]: 😐 😑 😕 😒 i'm not sure, but i'm glad to see what you did!
[User]: you're boring.
[RL - Model avg Psi]: 😴 ✋ 😒 why would you say this? what's up to you?!?!?!?!!!!!!!!!!!
[User]: i'm sorry! I didn't mean to offend you.
[RL - Model avg Psi]: 😔 😞 i'm not sure, and i think i'm going to do this, and i'm sure you'll go!!
[User]: i will go, you're being weird!
[RL - Model avg Psi]: 😠 👊 we are the same, we don't understand that, sir.
[User]: we are not the same. i'm a human and you're a bot.
[RL - Model avg Psi]: 💓 ✌️ that's really nice to me too.
>

Figure 4.18: Conversation with RL models on `https://neural.chat`.

recruited for interactive evaluation. Although the minimum required number of turns is 3, the average number of responses per conversation of participants varied between 3.00-10.58 turns with the average at 5.43 turns. Table 4.10 summarizes the number of ratings collected for each model. The average rating each annotator gave differed significantly between annotators. As a result, we also computed scores for interactive evaluation after normalizing each annotator's scores. We restricted ratings down to only annotators who completed 10 or more ratings which left 301 ratings. Similar to table 4.2, the mean ratings for EI (Emotion+Infersent) models were higher than the mean ratings for the baseline models.

	Cornell			Reddit		
	HRED	VHRED	VHCR	HRED	VHRED	VHCR
Baseline	55	46	53	55	36	39
EI	49	39	42	56	44	52

Table 4.10: Summary table of ratings collected per model.

4.5.6 *Website server setup and configuration*

The server was hosted on a Google Cloud Platform virtual instance with 64GB of RAM and a NVIDIA Tesla P100 graphics card. The backend was a Django program being served by NGINX and uWSGI. For simplicity, we opted to have the Django process import the chatbots into the same Python process as Django, rather than have the two connect to each other via other means such as sockets. This configuration decreased development time and increased reliability, but it would need to be revisited if the server needed to scale several orders of magnitude past what was required for this study. The current configuration was still able to support hundreds of simultaneous users and host more than 30 bots concurrently.

The chatbots were kept in a separate project from the Django project and maintained separately from the server code. Each chatbot extended an abstract class that defined key methods for the Django program to use, and was registered to a globally accessible dictionary via a decorator. The Django project was provided the path to the Chatbots project in its PYTHONPATH, so it could import the dictionary in which all the chatbot objects had been registered and use that to dynamically determine which chatbots were available and to access them in its views.

It is important to note that the chatbots used PyCUDA, and PyCUDA does not work in a multiprocessing environment. Because of this, uWSGI needed to be configured to only have one python process and to disable any attempt at multiprocessing. Furthermore, the chatbots required substantial startup times, so all chatbots are kept in memory at all times in the Django process. In order to keep all the chatbots in memory concurrently, we needed a very high amount of RAM on our server and opted for a 64GB virtual instance, and a GPU with 16GB RAM. This combination of CUDA to run the chatbots on the GPU with a high amount of RAM to keep all bots in memory at the same time resulted in incredibly fast server response times, with effectively no increase in response time when using the bots in requests compared to requests that did not.

For further information and instructions on server configuration, please read the server documentation available at `https://github.com/asmadotgh/neural_chat_web`. We hope that this platform will allow others to host their own bots and evaluate them in an interactive setting.

4.5.7 Hyper-parameter tuning details

For the baseline models that were trained on the Cornell dataset, we used the parameters reported in (Serban et al., 2016, 2017b; Park et al., 2018) that achieved state-of-the-art results for HRED, VHRED, and VHCR models trained on the same dataset, respectively. For EI models, we compared a combination of values for encoder hidden size (400, 600, 800, 1250), decoder hidden size (400, 600, 800, 1250), context size (1000, 1250), embedding size (300, 400, 500), word drop (0, .25), sentence drop (0, .25), beam size (1, 5). Learning rate (.0001), dropout (.2) were fixed. Batch size 80 was used. If due to memory limitation the job was not successfully completed, batch size 64 was used. Additionally, we tuned the EI parameters, i.e., emotion weight (25, 150), infersent weight (25K, 30K, 50K, 100K), emotion sizes (64, 128, 256), infersent sizes (128, 1000, 2000, 4000). Due to limited computational resources, we were not able to run a grid search on the aforementioned values. Instead we used combinations of the parameters that heuristically were more viable.

For the models that were trained on the Reddit dataset, a set of properly tuned baseline parameters were non-existent. Thus, to ensure fair comparison, we used a similar approach for baseline and EI hyper-parameter tuning: We explored a combination of values for encoder hidden size (400, 600, 800, 1250), decoder hidden size (400, 600, 800, 1250), context size (1000, 1250), embedding size (300, 400, 500, 600), word drop (0, .25), sentence drop (0, .1, .25), and beam size (1, 5). Learning rate (.0001), dropout (.2) were fixed. Batch size 64 was used. If due to memory limitation the job was not successfully completed, batch size 32 was used. Due to limited computational resources, we were not able to run a grid search on all the aforementioned values. Instead we used combinations of the parameters that heuristically were more viable. To ensure fair comparison, any selected combination was tested for both baseline and EI models. Then, for EI models, we tuned the parameters that were solely relevant to the EI design, such as the weight of emotion and infersent term in the loss function and the size of the added discriminator networks: Emotion weight (25), infersent weight (25K, 50K, 100K), emotion sizes (64, 128, 256), infersent sizes (100, 128, 1000, 2000, 4000). See Table 4.11 for a summary of the final selected parameters.

RL models were trained for between 800 and 1000 batches of data, where the batch size was fixed at 32. Early stopping was used to

Dataset	**Version**	**Model**	**Batch size**	**Dropout**	**Decoder hidden size**	**Encoder hidden size**	**Context size**	**Embedding size**	**Word drop**	**Sentence drop**	**Beam size**	**Emotion weight**	**Emotion discriminator layer size**	**Infersent weight**	**Infersent discriminator layer size**
Cornell	Baseline	HRED	80	.2	400	400	1000	300	.0	.0	5	-	-	-	-
		VHRED	80	.0	1000	1000	1000	400	.25	.0	5	-	-	-	-
		VHCR	80	.2	1000	1000	1000	500	.25	.25	5	-	-	-	-
	EI	HRED	64	.2	1000	1000	1000	500	.0	.0	1	25	128	100K	4000
		VHRED	80	.2	1250	1250	1000	600	.0	.0	1	25	128	30K	128
		VHCR	32	.2	1000	1000	1250	600	.0	.0	1	25	128	25K	4000
Reddit	Baseline	HRED	64	.2	1000	1000	1000	500	.0	.0	1	-	-	-	-
		VHRED	32	.2	1250	1250	1000	600	.0	.0	1	-	-	-	-
		VHCR	32	.2	1000	1000	1250	600	.0	.25	1	-	-	-	-
	EI	HRED	64	.2	1000	1000	1000	500	.0	.0	1	25	128	25K	2000
		VHRED	32	.2	1250	1250	1250	600	.0	.0	1	25	128	100K	4000
		VHCR	32	.2	1000	1000	1250	600	.0	.0	1	25	128	100K	4000

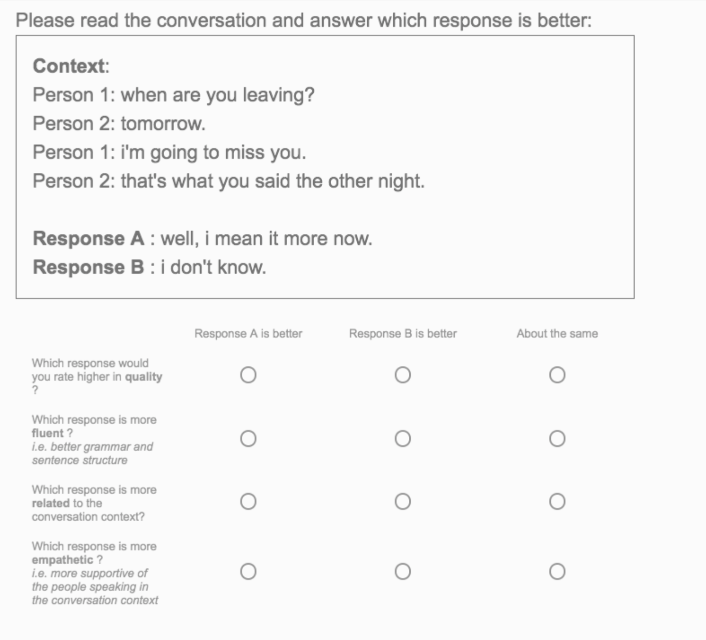
Table 4.11: Hyper-parameters used for different models.

determine the number of training iterations of the best checkpoint. All other hyperparameters were shared between RL models, and were as follows: discount $\gamma = 0.5$, weight placed on RL reward vs. KL-divergence term $c = 2$, number of Monte Carlo samples of the Target Q-network $M = 5$, target network update rate $\alpha = .005$, learning rate $r = .0001$. We used a smooth $L1$ loss function to approximate the Q-values, and clipped gradients at a value of 1.0.

The underlying parameters of the VHRED model were as follows: Context RNN hidden size $=$ 1000, decoder hidden size $=$ 1250, encoder hidden size $=$ 1250, z embedding size $=$ 600, gradient clip $=$ 1.0, dropout $d = 0.2$. The maximum conversation length was fixed at 5 utterances (context from more than 5 utterances ago was discarded), and the maximum sentence length was 30 tokens. There were 2 additional feedforward EI prediction layers of size 128, which used ReLu activation.

4.5.8 *One-turn evaluation experiment details*

Figure 4.19 shows the one-turn evaluation interface that Mechanical Turk workers used.



Please read the conversation and answer which response is better:

Context:
Person 1: when are you leaving?
Person 2: tomorrow.
Person 1: i'm going to miss you.
Person 2: that's what you said the other night.

Response A : well, i mean it more now.
Response B : i don't know.

	Response A is better	Response B is better	About the same
Which response would you rate higher in **quality** ?	○	○	○
Which response is more **fluent** ? *i.e. better grammar and sentence structure*	○	○	○
Which response is more **related** to the conversation context?	○	○	○
Which response is more **empathetic** ? *i.e. more supportive of the people speaking in the conversation context*	○	○	○

Figure 4.19: One-turn evaluation interface crowdworkers see.

4.5.9 *Additional results*

4.5.9.1 *Ablated EI models*

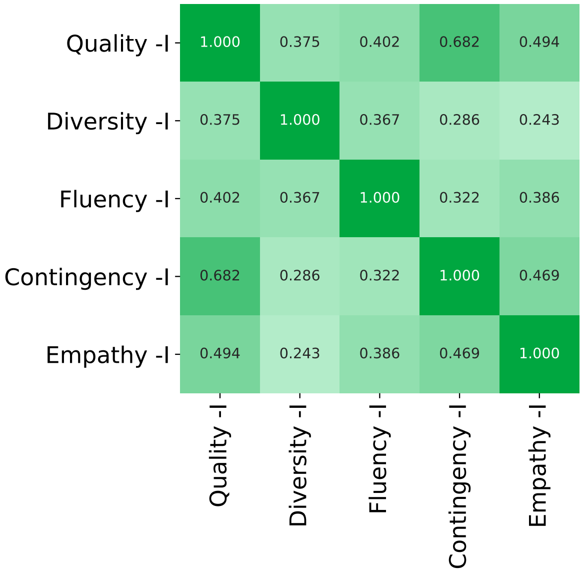
We conducted additional evaluations of ablations of our EI models, to determine whether emotion or infersent regularization provided the most benefit. The results in Table 4.12 reveal that this depends on the dataset and the model in question. We also checked whether simply appending the emotion and infersent embedding of an utterance to the top level of the hierarchy could provide the same benefit as knowledge distillation, even though this would require retaining copies of the DeepMoji and Infersent models, and would be more computationally expensive at inference time. Table 4.12 reveals that the *input-only* models do not out-perform the knowledge-distillation EI models on automatic metrics.

		Cornell					Reddit				
Model	Version	PPL	KL	Avg	Ext	Grd	PPL	KL	Avg	Ext	Grd
HRED	baseline	52.311	-	.471	.329	.331	41.730	-	.649	.394	.474
	input only	47.911	-	.549	.381	.392	41.227	-	.644	.395	.469
	EI_{emo}	48.619	-	**.562**	.359	**.416**	47.395	-	.541	.310	.371
	EI_{inf}	47.988	-	**.562**	.381	.405	**41.083**	-	.646	.394	.472
	EI	**47.636**	-	.560	**.383**	.400	41.245	-	**.651**	**.398**	**.482**
VHRED	baseline	**49.414**	.264	.539	.352	.395	36.240	.188	.635	.383	.464
	input only	49.819	.442	.543	.353	.393	40.248	.312	.630	.377	.456
	EI_{emo}	51.346	.636	**.552**	**.358**	**.401**	36.212	.199	.631	.380	.458
	EI_{inf}	52.143	**.702**	.539	.346	.392	36.518	**.222**	**.637**	.381	.463
	EI	50.526	.517	.545	.355	.394	**35.510**	.167	.636	**.392**	**.465**
VHCR	baseline	61.000	.562	.532	.345	.382	**36.736**	.267	.619	.371	.448
	input only	50.966	.558	.531	.344	.382	37.342	**.287**	.608	.365	.431
	EI_{emo}	52.407	.590	.585	**.374**	.442	37.449	.254	.619	.366	.444
	EI_{inf}	53.085	**.575**	.544	.356	.390	37.109	.199	.629	.378	.457
	EI	**49.243**	.475	**.588**	.369	**.444**	37.198	.231	**.639**	**.394**	**.469**

Table 4.12: Automatic metrics computed on ablations of the EI models, trained with distillation from only the emotion recognition model (EI_{emo}), the infersent model (EI_{inf}), or receiving emotion and infersent only as input, without knowledge distillation (*input-only*). Whether emotion or semantics provides the most benefit depends on the dataset and the model.

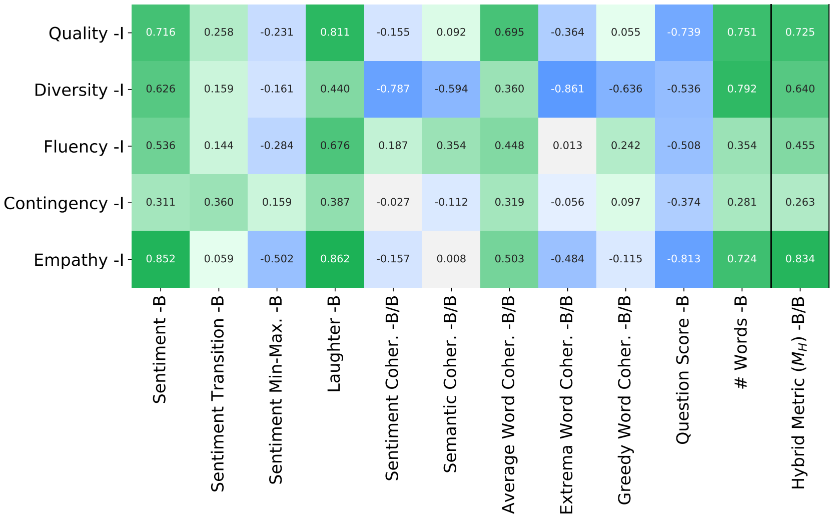
4.5.9.2 *Human interactive ratings correlation table*

Figure 4.20 provides detailed information about different aspects of interactive human ratings. We observe that quality is highly correlated with other aspects of the conversation. Specifically, it is most strongly correlated with contingency, which further highlights the importance of semantic metrics of bot-generated responses in a good quality conversation. It also has high correlation with empathy that could better be captured by sentiment metrics.



	Quality -I	Diversity -I	Fluency -I	Contingency -I	Empathy -I
Quality -I	1.000	0.375	0.402	0.682	0.494
Diversity -I	0.375	1.000	0.367	0.286	0.243
Fluency -I	0.402	0.367	1.000	0.322	0.386
Contingency -I	0.682	0.286	0.322	1.000	0.469
Empathy -I	0.494	0.243	0.386	0.469	1.000

Figure 4.20: Correlation matrix showing the relationships between different aspects of interactive human evaluation. We observe a strong correlation across these human ratings.



	Sentiment -B	Sentiment Transition -B	Sentiment Min-Max. -B	Laughter -B	Sentiment Coher. -B/B	Semantic Coher. -B/B	Average Word Coher. -B/B	Extrema Word Coher. -B/B	Greedy Word Coher. -B/B	Question Score -B	# Words -B	Hybrid Metric (M_H) -B/B
Quality -I	0.716	0.258	-0.231	0.811	-0.155	0.092	0.695	-0.364	0.055	-0.739	0.751	0.725
Diversity -I	0.626	0.159	-0.161	0.440	-0.787	-0.594	0.360	-0.861	-0.636	-0.536	0.792	0.640
Fluency -I	0.536	0.144	-0.284	0.676	0.187	0.354	0.448	0.013	0.242	-0.508	0.354	0.455
Contingency -I	0.311	0.360	0.159	0.387	-0.027	-0.112	0.319	-0.056	0.097	-0.374	0.281	0.263
Empathy -I	0.852	0.059	-0.502	0.862	-0.157	0.008	0.503	-0.484	-0.115	-0.813	0.724	0.834

Figure 4.21: Correlation matrix showing the relationships between different automated metrics on self-play trajectories and interactive human ratings aggregated on the bot-level. **Postfixes:** -I: Interactive human evaluation, -B: Calculated on bot response, -B/B: Metric applied to self-play on two consecutive bot generated utterances when the bot converses with itself.

		Cornell			Reddit		
Model	Metric	Wins %	Losses %	Ties %	Wins %	Losses %	Ties %
HRED	quality	**40.8** ± 4.9	24.5 ± 4.9	34.8 ± 9.2	**31.3** ± 5.2	29.5 ± 6.6	39.3 ± 10.7
	fluency	10.3 ± 4.4	**17.3** ± 4.1	72.5 ± 8.1	**22.8** ± 5.3	20.0 ± 7.1	57.3 ± 11.2
	relatedness	36.3 ± 6.5	28.7 ± 4.8	35.0 ± 7.9	**34.3** ± 2.8	30.3 ± 7.8	35.5 ± 9.7
	empathy	**37.8** ± 7.2	24.5 ± 5.6	37.8 ± 8.9	**32.5** ± 3.4	31.2 ± 5.9	36.3 ± 8.0
VHRED	quality	**36.9** ± 4.7	36.6 ± 5.6	26.6 ± 6.9	**39.0** ± 7.0	34.0 ± 5.3	27.0 ± 8.9
	fluency	23.4 ± 9.6	**27.7** ± 8.3	48.9 ± 16.3	**29.0** ± 13.6	23.3 ± 9.3	47.7 ± 21.6
	relatedness	**37.4** ± 5.4	33.1 ± 7.2	29.7 ± 9.6	**38.3** ± 5.6	33.0 ± 5.1	28.7 ± 9.0
	empathy	**36.6** ± 9.4	34.0 ± 8.4	29.4 ± 15.8	**34.7** ± 8.7	33.7 ± 6.7	31.7 ± 10.9
VHCR	quality	**33.0** ± 6.1	29.0 ± 5.4	38.0 ± 10.1	**33.7** ± 7.9	27.3 ± 3.3	39.0 ± 8.6
	fluency	13.5 ± 4.1	**25.5** ± 4.3	66.0 ± 7.7	**24.7** ± 7.2	18.3 ± 5.2	57.0 ± 10.2
	relatedness	**40.8** ± 4.8	26.8 ± 6.8	32.5 ± 10.5	28.3 ± 6.6	**31.3** ± 3.6	40.3 ± 8.4
	empathy	**32.8** ± 6.6	28.0 ± 7.8	39.3 ± 13.7	**30.3** ± 3.9	24.0 ± 4.6	45.7 ± 7.6

Table 4.13: Results from human single-turn evaluation for EI vs. Baseline models for HRED, VHRED, and VHCR models across quality, fluency, relatedness and empathy pairwise comparisons with 90% confidence intervals

4.5.9.3 *Self-play correlation table*

Figure 4.21 provides detailed information about the introduced metrics applied to self-play. We observe that several metrics also transfer to self-play trajectories. Inducing positive sentiment as measured by Sentiment and Laughter, and being able to generate longer sentences in self-play are associated with higher quality model ratings. It is worth mentioning that maintaining extreme similarity in sentiment or semantics or just asking questions in self-play conversation trajectories could backfire by reducing the diversity of generated responses, though applicable to interactive human data. Most importantly, our novel hybrid metric applied to self-play (M_H -B/B) is highly correlated with all human ratings of the dialog model.

4.5.9.4 *Detailed results of single-turn evaluation*

We replicated the one-turn evaluation found in previous work (Serban et al., 2017b; Park et al., 2018). We sampled conversation contexts from the test set of each corpus and generated samples by each model based on these contexts. After filtering by context length (> 10 tokens) and removing contexts which contain <unknown> tokens, we sampled 100 examples. We divided each set of 100 examples into two batches of 50 for annotators to rate. Annotators recruited through Amazon Mechanical Turk were first trained with an example question. Annotators must be in the United States and had to correctly answer all training questions to complete the task. Figure 4.19 shows the interface displayed to crowdworkers in the one-turn evaluation task. We asked annotators to select which sentence was better for quality, fluency, relatedness, and empathy. Table 4.13 summarizes the results for all 4 metrics and is an uncondensed version of table 4.4. One notable exception to the pattern of EI models winning is fluency; baseline models trained on the CORNELL corpus generated more fluency wins.

Noting the high disagreement between annotators in this task, we further examined the ambiguous examples in the human evaluation test set. We define an ambiguous example as a question where an equal number of annotators select the first sentence as better as the second sentence. If the two examples were similar, annotators would select the "tied" option. An equal number of selections for each answer as the winner indicates a disagreement in perception. Table 4.14 summarizes the number of ambiguous examples per model and metric out of 100 in total for each box. After removing these ambiguous example from calculating wins, losses and ties, the results are similar to table 4.2. The number of ambiguous samples further highlights the noisy and unreliable nature of single-turn evaluation.

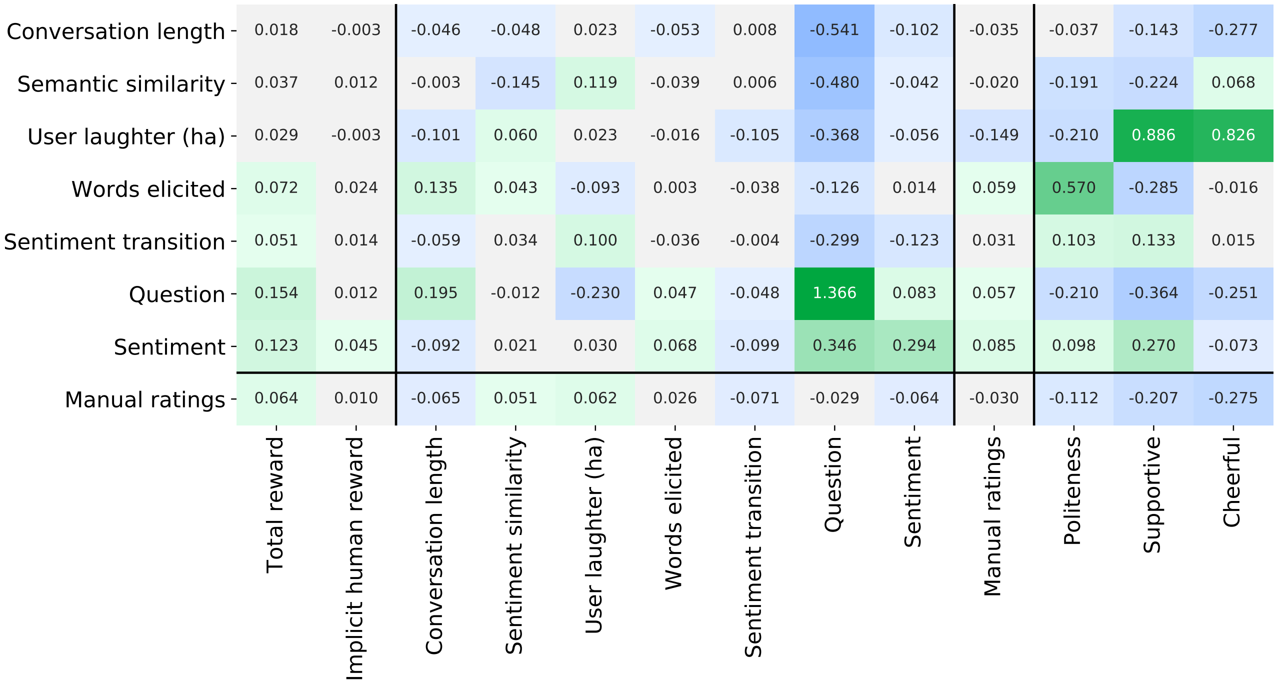
	Cornell			Reddit		
	HRED	VHRED	VHCR	HRED	VHRED	VHCR
Quality	12	13	15	26	15	9
Fluency	4	10	10	12	20	6
Relatedness	11	12	10	15	13	7
Empathy	16	9	12	14	17	7

Table 4.14: Count of ambiguous examples in human one-turn evaluation.

4.5.9.5 Additional RL results

Figure 4.22 shows the normalized[6] reward scores obtained by bots trained with respect to different rewards. While some bots (such as those trained to ask questions or elicit positive sentiment) effectively generalize to new users, we see that others (e.g. words elicited) are not actually able to best elicit those responses in the wild. We hypothesize this is because the relatively small size of batch date we were able to collect ($\approx 14,000$ utterances) does not give these bots enough information about how to elicit long responses from users.

[6] Using *z*-score normalization



	Total reward	Implicit human reward	Conversation length	Sentiment similarity	User laughter (ha)	Words elicited	Sentiment transition	Question	Sentiment	Manual ratings	Politeness	Supportive	Cheerful
Conversation length	0.018	-0.003	-0.046	-0.048	0.023	-0.053	0.008	-0.541	-0.102	-0.035	-0.037	-0.143	-0.277
Semantic similarity	0.037	0.012	-0.003	-0.145	0.119	-0.039	0.006	-0.480	-0.042	-0.020	-0.191	-0.224	0.068
User laughter (ha)	0.029	-0.003	-0.101	0.060	0.023	-0.016	-0.105	-0.368	-0.056	-0.149	-0.210	0.886	0.826
Words elicited	0.072	0.024	0.135	0.043	-0.093	0.003	-0.038	-0.126	0.014	0.059	0.570	-0.285	-0.016
Sentiment transition	0.051	0.014	-0.059	0.034	0.100	-0.036	-0.004	-0.299	-0.123	0.031	0.103	0.133	0.015
Question	0.154	0.012	0.195	-0.012	-0.230	0.047	-0.048	1.366	0.083	0.057	-0.210	-0.364	-0.251
Sentiment	0.123	0.045	-0.092	0.021	0.030	0.068	-0.099	0.346	0.294	0.085	0.098	0.270	-0.073
Manual ratings	0.064	0.010	-0.065	0.051	0.062	0.026	-0.071	-0.029	-0.064	-0.030	-0.112	-0.207	-0.275

Figure 4.22: Normalized reward scores obtained by models trained with respect to different rewards.

5 Learning from human facial expressions for generating sketches

5.1 *Introduction* 192
5.2 *Related work* 194
5.3 *Methods* 195
5.3.1 *Study design* 195
5.3.2 *Machine learning techniques* 196
5.4 *Experiments* 197
5.5 *Results* 198
5.5.1 *Facial expression analysis* 198
5.5.2 *Machine learning results* 200
5.6 *Conclusions and future work* 201
5.7 *Statement of contributions* 202

A known deficit of modern machine learning (ML) and deep learning (DL) methodology is that models must be carefully fine-tuned in order to solve a particular task. Most algorithms cannot generalize well to even highly similar tasks, let alone exhibit signs of artificial general intelligence (AGI). To address this problem, researchers have explored developing loss functions that act as intrinsic motivators that could drive an ML or DL agent to learn across a number of domains. Once again, this chapter argues that social interaction can be an important and useful intrinsic motivator, which can help agents adapt quickly and flexible to new tasks. We posit that making an AI agent aware of implicit social feedback from humans can allow for faster learning of more generalizable and useful representations, and could potentially impact AI safety. To support this claim, we collect social feedback in the form of facial expression reactions to samples from Sketch RNN, an LSTM-based variational autoencoder (VAE) designed to produce sketch drawings. We use a Latent Constraints

GAN (LC-GAN) to learn from the facial feedback of a small group of viewers, by optimizing the model to produce sketches that it predicts will lead to more positive facial expressions. We show in multiple independent evaluations that the model trained with facial feedback produced sketches that are more highly rated, and are associated with significantly more positive facial expressions. Thus, we establish that implicit social feedback can improve the output of a deep learning model.

5.1 Introduction

Despite the recent rapid and compelling progress in ML and DL, modern AI is still remarkably far from approximating the intelligence of even simple animals. A notable deficit is the degree of explicit supervision required in order to learn, either through labeled samples or well-defined external rewards such as points in a game. The limited scope of such supervision will not enable the development of a generally intelligent AI.

For this reason, some researchers have focused on intrinsic motivators, inherent drives that cause the agent to learn representations that are useful across a variety of tasks and environments. Examples include curiosity (a drive for novelty) (Pathak et al., 2017), and empowerment (a drive for the ability to manipulate the environment) (Capdepuy et al., 2007). However, so far this research has overlooked an important intrinsic motivator for humans: the drive for positive social interactions.

We argue that making an AI agent intrinsically motivated to obtain a positive social reaction from humans in its environment is an important new research direction. Specifically, the agent should be able to recognize implicit feedback from humans in the form of facial expressions, body language, or tone in voice and text, and optimize for actions that appear to please humans as measured through these signals. Such feedback could be used in a wide range of contexts where people interact with ML systems.

The representations learned by such an agent are more likely to capture dimensions of the task that are relevant to human satisfaction. This has meaningful implications for questions of AI safety; an AI agent motivated by satisfaction expressed by humans will be less likely to take actions against human interest. Such an agent will also be better suited to perform tasks which already involve AI. Imagine if a home assistant could sense when a user responds with an angry or frustrated tone and this acted as a negative incentive, training the algorithm not to repeat the action that led to the user's frustration? Rather than requiring the user to manually train the device, it could learn quickly through passive sensing of the user's emotional state, leading to a more immediately satisfying experience for the user. Finally, some machine learning problems—including the one under investigation in this chapter—cannot be solved without human feedback; when the objective function is human aesthetic preference, it cannot be approximated without human input.

Social awareness may be a key component of AGI. There is substantial evidence that emotion recognition, which is critical for empathy and successful social interaction, plays an influential role in cognitive development in humans (Kujawa et al., 2014). According to Social Learning Theory (Bandura and Walters, 1977), observing the attitudes and behaviors of others is a central component of how humans learn both intelligent behavior and how to adapt to new situations. It has been argued that social learning is responsible for the rapid cultural evolution of the human species (van Schaik and Burkart, 2011). Given the importance of cultural evolution to humans' technological success, endowing a deep learning agent with the ability to perceive and benefit from this socially exchanged cultural knowledge could allow it to rapidly develop more generalizable knowledge representations.

In this work we demonstrate the utility of learning through implicit social feedback via an experiment in which samples generated by a deep learning model are presented to people, and their facial expression response is detected. The model is Sketch RNN (Ha and Eck, 2017), an LSTM-based VAE with a Mixture Density Network output, designed to produce sketch drawings. Using a newly developed technique known as Latent Constraints (Engel et al., 2017), we train a Generative Adversarial Network (GAN) to produce VAE embedding vectors that, when decoded by Sketch RNN, are more likely to produce drawings that lead to positive facial expressions such as smiling. In a rigorous, double-blind evaluation, we show that samples from the social feedback model generate statistically significantly better affective responses than the prior, and are consistently rated as more preferred by human judges. Thus, this experiment is a first step in demonstrating that deep learning models are able to improve in quality as a result of learning from implicit social feedback.

5.2 *Related work*

Many affective computing papers have addressed how to automatically detect facial expressions (e.g. Senechal et al. (2015)). A comprehensive review of this work is out of scope for this chapter. We instead build on this work by assuming that an accurate facial expression detector is already available, and asking what can be learned using this facial feedback.

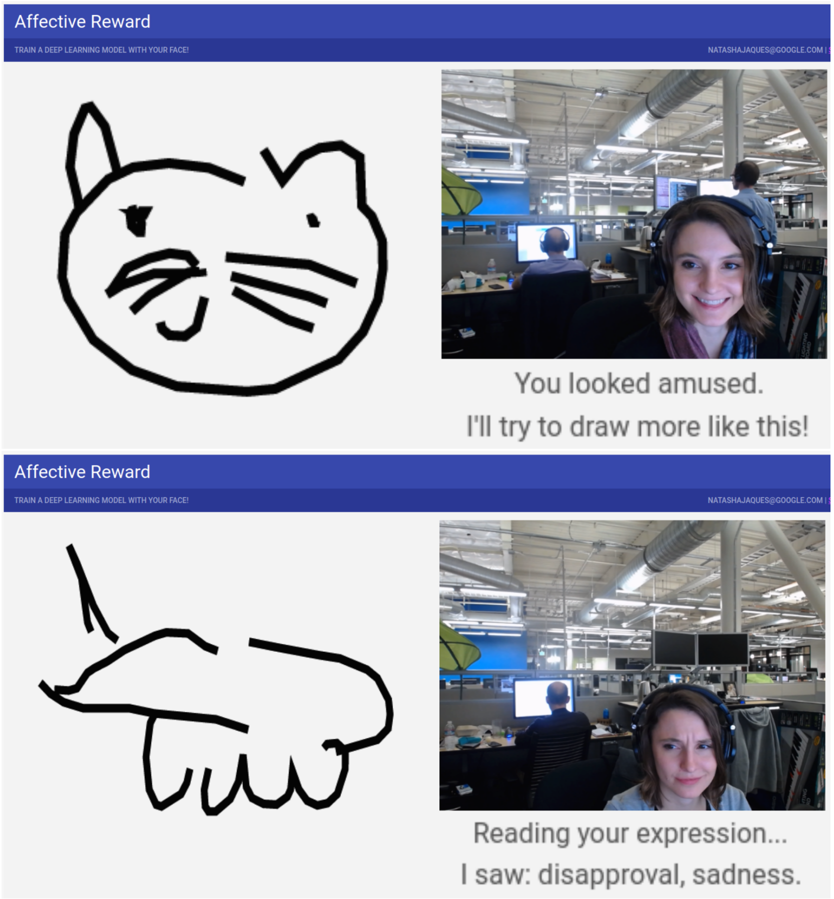
Previous work has attempted to train ML and DL models to approximate human preferences. For example, Knox and Stone (2009) ask users to press a button to teach a reinforcement learning (RL) model to play *Mountain Car*, and model human reaction latencies as a Gamma distribution in order to distribute the reward appropriately over past time steps. A more recent work attempts to train a deep learning model from human preferences, by first training an approximator of human button presses using supervised learning, and then using this to train an RL model (Christiano et al., 2017). However, both of these approaches require the human to provide explicit supervision by manually entering feedback. In contrast, our approach enables learning from implicit social cues that can be obtained ubiquitously, through awareness of the non-verbal reactions people naturally provide. Essentially, we obtain human-in-the-loop training without additional human effort.

The closest work to our own of which we are aware is an approach that used valence and engagement, detected via facial expressions, as a reward function in a Q-learning framework (Gordon et al., 2016). The goal of the project was to allow an intelligent tutoring system to adapt its behavior so that children would remain engaged while using the system. While this is an excellent example of learning from implicit social feedback, the goals of this paper are quite distinct from our own. We believe we are the first authors to use implicit social feedback to improve a generative deep learning model. Our model attempts to learn to improve its ability to produce creative content by observing the implicit responses it receives from human judges. This process could be considered analogous to a human artist fine-tuning her work after she observes critics' nonverbal reactions.

5.3 Methods

5.3.1 Study design

To gather social feedback, we focused on facial expression recognition, since this is currently one of the most reliable and accurate ways to detect social signals (Senechal et al., 2015). The facial expression detector employed for this project is a pre-trained convolutional network trained to detect common facial expressions.


Affective Reward
TRAIN A DEEP LEARNING MODEL WITH YOUR FACE!
NATASHAJAQUES@GOOGLE.COM

You looked amused.
I'll try to draw more like this!

Affective Reward
TRAIN A DEEP LEARNING MODEL WITH YOUR FACE!
NATASHAJAQUES@GOOGLE.COM

Reading your expression...
I saw: disapproval, sadness.

Figure 5.1: The web app was designed to indicate to users how the model was interpreting their facial expressions in order to learn how to sketch better.

To obtain facial feedback at scale, we built a web app that serves samples from a deep learning model while recording the user's facial expressions with a webcam. Figure 5.1 shows examples of the interface, which was carefully designed in conjunction with a User Experience team at Google (including James Tolentino, Ira Blossom, Adrien Baranes, Rebecca Salois, and Josh Lovejoy) in order to elicit facial expressions from the user, and clearly communicate how they were being used and interpreted. The user's image was collected via their webcam and fed into the facial-expression detection network to compute the intensities of common expressions, including amusement, contentment, surprise, sadness, and concentration. Due to the degree of inter-individual variation in users' resting facial expression, these intensities were normalized against each user's average expression to produce value vectors v. The app is also capable of collecting Likert-scale ratings of sketch quality and asking users to choose which of two sketches they prefer. These mechanisms were used to collect evaluation data. The app can be viewed at `https://facial-feedback-for-ai.appspot.com/`.

To test the hypothesis that facial feedback can improve the outputs of a deep learning model, we sought a model for which the outputs were likely to generate a natural facial expression response. We chose Sketch RNN (Ha and Eck, 2017), a model which generates sequences of strokes that form a sketched image of a common object, vehicle, or animal (see Figure 5.6). Such sketches were determined to elicit facial responses in initial tests.

5.3.2 *Machine learning techniques*

Sketch RNN is a VAE that was trained in an unsupervised manner on a large corpus on human sketch data collected via *Quick, Draw!*[1]. The sketches are represented as sequences of coordinates that represent the points where the pen is placed during sketching. The architecture of Sketch RNN comprises: a) a bidirectional LSTM encoder that projects each input sketch into a latent embedding vector z, b) an LSTM decoder which takes z as input and generates a sequence of parameters for c) a Gaussian Mixture Model that generates the (x, y) coordinates of the tip of the pen during each stroke. This Mixture Density Network (MDN) approach is similar to prior work on handwriting generation (Graves, 2013).

The design of Sketch RNN provides important benefits that facilitate optimizing the model with facial feedback. First, due to the variational constraint, it is straightforward to sample a latent vector $z \sim \mathcal{N}(0, I)$ and feed this into the Sketch RNN decoder to produce a recognizable sketch. Second, the latent embeddings z learned by Sketch RNN provide a clean, compressed representation of sketch drawings. These features allowed us to apply a newly developed technique known as Latent Constraints (Engel et al., 2017) in order to learn to produce sketches likely to lead to positive facial expressions.

---

[1] `https://quickdraw.withgoogle.com/`

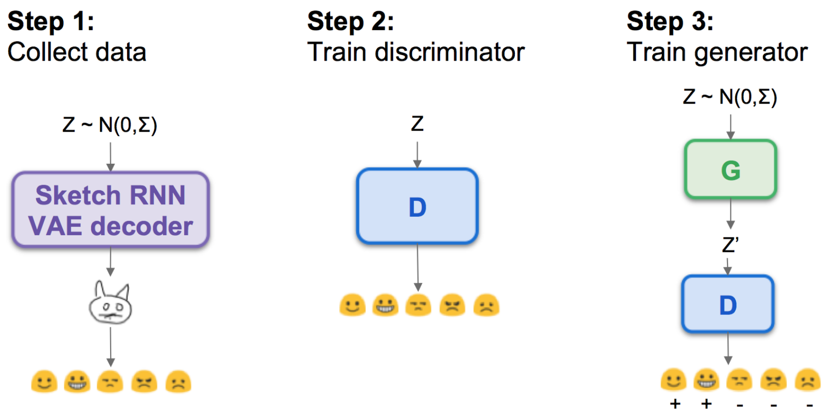
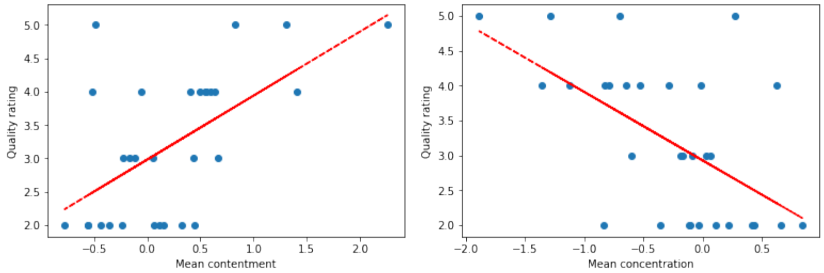


Figure 5.2: Steps involved in training the LC-GAN facial feedback model.

The latent constraints GAN (LC-GAN) is a GAN applied to the latent embedding space of a VAE. The steps of training this model to use facial feedback are shown in Figure 5.2. We first sample a number of z vectors from the VAE prior ($\mathcal{N}(0, I)$), and feed these into the Sketch RNN decoder to obtain sketches. These sketches are shown to users, and the intensity of their facial expression responses is recorded; we refer to these intensities as the *value* of a sketch, v. The z vectors are then used as input to a discriminator $D(z) \rightarrow v$, which is trained to estimate the value v of different regions of the latent space; for example, which regions decode to sketches that produced the highest intensity of smiles. A generator $G(z) \rightarrow z'$ is then trained to convert a randomly sampled z into a modified z' that produces a higher v. In fact, the generator uses a gating mechanism to control how heavily the original z is modified. The generator loss is $\mathcal{L}_G = -\log D(z')$. Because the Sketch RNN latent space is a compressed, 128-dimensional, robust representation, the discriminator is able to learn a value function on z even with relatively small sample sizes.

5.4 *Experiments*

Data collection for the experiments was conducted in four phases. In the initial phase, we ran a pilot study on 7 users who viewed a total of 30 sketches, in which we collected both facial expressions and Likert-scale ratings of sketch quality. Then, we used the webapp to collect facial reactions from 28 users to a total of 334 sketches, recording the embedding vector z for each sketch. These (z, v) pairs were used to train the LC-GAN. Finally, two phases of data collection were used to evaluate the model. Both involved rigorous, double-blind experiments in which we randomly generated hundreds of samples from both the facial feedback and baseline models, and displayed them in random order to users "in the wild", using their personal webcams, without experimenter supervision. The first evaluation sought to establish that sketches from the LC-GAN are able to elicit more positive facial expressions than the original Sketch RNN. For this test, we obtained evaluation data from 76 users, spanning 536 sketches. The second evaluation asked users to rate which of two sketches they preferred; we collected 4,692 ratings from 79 users.



(a) Contentment, r=-.58, p <.001

(b) Concentration, r=.58, p <.001

Figure 5.3: Scatterplots showing that human ratings of quality are highly correlated with their facial expressions.

5.5 *Results*

5.5.1 *Facial expression analysis*

Optimizing for user preference requires knowing which facial expressions indicate that the user likes a sketch. Therefore, we used the pilot study data to assess how users' ratings of sketch quality related to their facial expressions. We found significant positive correlations with contentment and amusement (smiling), and significantly negative correlations with sadness and concentration (frowning), as shown in Table 5.1; examples are shown in Figure 5.3. Notably, these results indicate that implicit facial feedback carries an informative signal about the user's preferences.

Emotion metric	r	p
Contentment	.582	.001
Amusement	.546	.002
Concentration	-.576	.001
Sadness	-.405	.026

Table 5.1: Significant correlations between facial expressions and quality ratings.

However, learning from facial feedback still represents a challenging problem; there is a high degree of inter-individual variability, the meaning of facial expressions may be extremely context dependent, and the data can be remarkably noisy. In addition to noise introduced through inaccuracies in the detector, there are many confounding reasons that may cause a person to make a given facial expression.

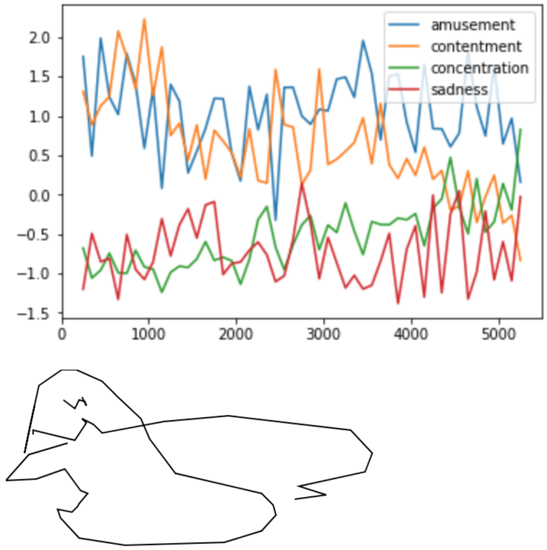


Figure 5.4: The user expresses contentment and amusement at a scribbled sketch, showing the challenging level of noise present in the human facial expression feedback.

For example, Hoque and Picard (2011) found that users tend to smile when they are frustrated. In our case, we noticed that users tend to smile simply at the concept of an AI making drawings, or even at their own face as shown to them via the webcam feed. This can lead to highly misleading interactions; Figure 5.4 shows an example in which the user smiles profusely at a drawing that is no better than a scribble.

Finally, the difficulty of modeling user preference through facial expressions is enhanced by the non-stationarity of the data. We found that users' facial expressions tended to change over repeated interactions with the system. There were significant relationships between the number of previous sketches viewed by the user and the user's average sadness ($r(751) = 0.248, p < .001$) and concentration $r(751) = -0.158, p < .001$). Thus, the meaning of an intense expression of concentration may change depending on when it occurs in the user's interaction with the app.

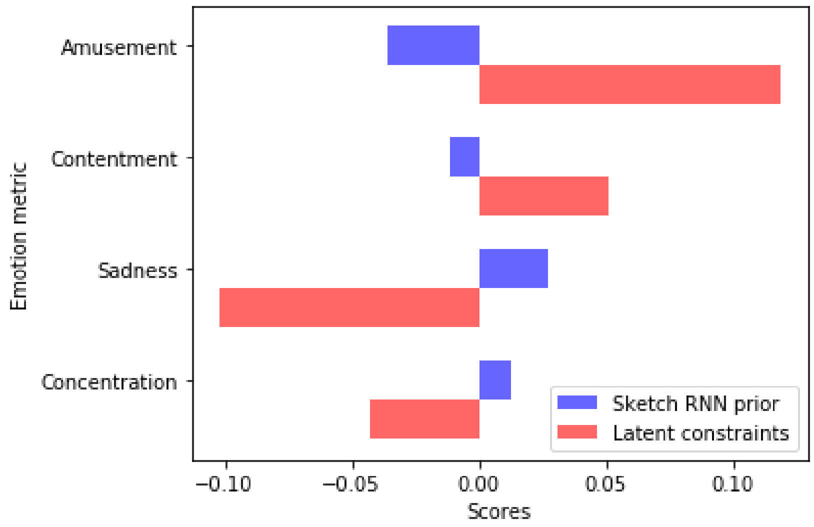


Figure 5.5: Sketches from the LC-GAN lead to significantly more amused expressions and significantly less sadness.

5.5.2 *Machine learning results*

In spite of the noise and non-stationarity inherent in the data, we found that the LC-GAN was able to use the facial feedback to learn to produce significantly higher quality sketches. Given the direction of the relationships discovered between facial expressions and quality, we trained the LC-GAN to maximize amusement and contentment, and minimize concentration and sadness. Although relatively little data was collected (63-69 samples per sketch class), the LC-GAN was able to effectively optimize for more pleasing sketches. Figure 5.6 shows the difference between samples produced with the Sketch RNN prior and the LC-GAN. The LC-GAN appears to have learned that people smile more and frown less for cats with larger, smiling faces with whiskers. Similarly, the quality of crab and rhinoceros sketches generated by the LC-GAN appears to be consistently higher. For example, the original rhinoceros model often produced sketches that did not resemble a rhinoceros, or were no better than scribbles. After training with a small amount of facial feedback, the LC-GAN model consistently produces more realistic drawings.

The evaluation data revealed that humans found sketches from the LC-GAN model to be significantly better. In the first experiment, we were able to support the hypothesis that sketches from the LC-GAN model generate significantly more positive facial expressions than the original Sketch RNN. Figure 5.5 shows the results of this evaluation, indicating that all of the facial expression metrics improved in the

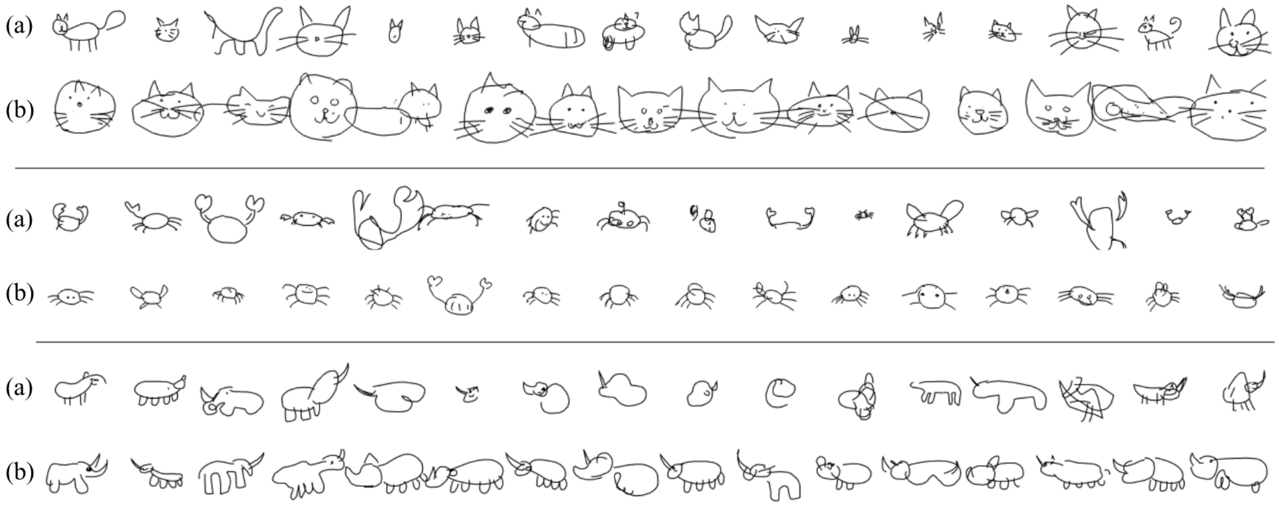


Figure 5.6: Samples drawn randomly from the cat, crab, and rhinoceros sketch classes, produced by (a) the original Sketch RNN, and (b) the LC-GAN trained on a small amount of social feedback.

expected direction under the LC-GAN. Two of the metrics reached statistical significance: mean amusement, $t(535) = 2.31, p < .05$, and mean sadness, $t(535) = -2.01, p < .05$. In the second part of the evaluation, we tested the hypothesis that humans actually rate the quality of the LC-GAN sketches as higher. Users reported preferred the LC-GAN 2843 times, as opposed to 1770 for the original Sketch RNN, a significant improvement as shown in a Binomial test, $p < .0001$.

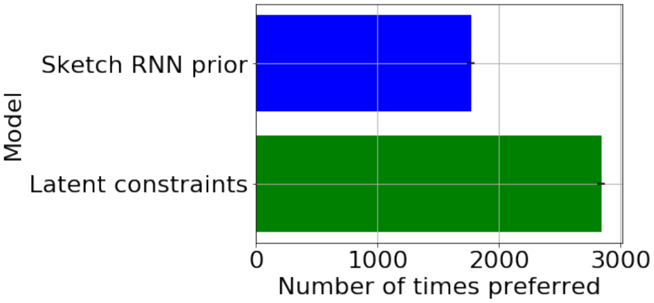


Figure 5.7: Sketches from the LC-GAN are consistently preferred over sketches from the original model. Error bars show the variance of the estimated binomial distribution.

5.6 *Conclusions and future work*

We have demonstrated that implicit social feedback in the form of facial expressions not only can reflect user preference, but also can significantly improve the performance of a deep learning model.

There are many ways to enhance and extend this work. For example, we could use an RL framework to improve the model's ability to draw based on facial feedback. Further, our current model makes no use of the evolving temporal dynamics of the collected facial expressions, instead relying on an average intensity over viewing the sketch. A more sophisticated system could use the alignment between the process of drawing the sketch and the user's expressions over time to gain better temporal credit assignment in an RL framework.

5.7 Statement of contributions

This work initially began as an internship project at Google Brain under the supervision of Douglas Eck. I proposed learning from human facial expression feedback to improve a deep learning model. In close consultation with Fred Bertsch, I determined that the model most likely to elicit facial expression responses was David Ha's Sketch RNN. I built a website that was able to serve samples of sketch RNN outputs to humans and collect their facial expression response. We worked with a Human Centered Machine Learning (HCML) team comprising James Tolentino, Ira Blossom, Adrien Baranes, Rebecca Salois, and Josh Lovejoy to conduct a pilot study using the website and analyze the results. I modified Latent Constraints code provided by Jesse Engel and used it to train Sketch RNN in response to facial feedback data. After leaving Google, Rosalind Picard provided advice on how to extend the project, and Jennifer McCleary re-implemented the website I had built in open source using Affectiva, so that I could conduct additional human evaluation experiments with the Latent Constraints models.

6 Modeling bonding in human conversations

6.1 Introduction 204
6.2 Related work 205
6.3 User study 206
6.4 Methods 208
6.4.1 Facial expression detection 208
6.4.2 Skeletal joint extraction 208
6.4.3 Machine learning classification 209
6.5 Results 210
6.5.1 Reliability of the bonding scale 210
6.5.2 Designing an agent to promote bonding 211
6.5.3 Qualitative analysis of non-verbal cues 212
6.5.4 Quantitative analysis of non-verbal cues 213
6.5.5 Predicting bonding in novel conversations. 217
6.6 Conclusions and future work 218
6.7 Statement of contributions 219

If we want AI to learn from implicit human signals, we must first create systems that accurately detect those signals. This chapter investigates how to predict whether two humans involved in a conversation with each other are experiencing a feeling of *bonding* or chemistry, based on their facial expressions and body language. This could allow an intelligent agent to predict whether it is bonding with its user, and take actions to enhance this effect. To this aim, we investigate how the agent could convey appropriate facial expression and body language responses to foster bonding. Observational data, including video and Kinect recordings, are collected from a series of naturalistic conversations, and a reliable measure of bonding is adapted and verified. A qualitative and quantitative analysis is conducted to determine the non-verbal cues that characterize both high and low bonding conversations. We then train a deep neural network classifier using one-minute segments of facial expression and body language data, and show that it is able to accurately predict bonding in novel conversations. Further, the effects of personality and dispositional attitudes on bonding are analyzed, and we find that attentiveness and excitement are more effective at promoting bonding than traits like attractiveness and humour.

6.1 Introduction

The most effective conversationalists do not simply smile, nod, and mirror their partner; instead, they are adept at sensing non-verbal cues and adapting to the other person's state. If an intelligent virtual agent (IVA) could be designed with this level of emotional intelligence, it could dynamically adapt its interaction style to the needs of the user. Such an endearing and empathetic IVA would have a wide range of applications, from intelligent tutoring, to human-robot interaction, to helping individuals who struggle with social interaction.

Many studies have probed how to make intelligent virtual agents (IVAs) more appealing to human users, by focusing on the aesthetic appeal of the characters (e.g. van Vugt et al. (2009)), their facial expressions (e.g. Wong and McGee (2012)), mirroring (e.g. Kahl and Kopp (2015)), and the contingency of their non-verbal responses (Gratch et al., 2006). Detailed models of bonding and rapport (Zhao et al., 2014), and interpersonal emotions in conversations (Butler, 2011), have also been developed. We contribute to this work by examining which dispositional attitudes and personality traits are most important to bonding and rapport. For example, if Agreeableness is important to rapport (as reported in Cuperman and Ickes (2009)), it may suggest that designing the responses of an IVA to appear more kind, polite, and non-confrontational would be beneficial (Carver and Scheier, 2011). The effect of personality on bonding is compared to that of traits like attractiveness and humour, to suggest which characteristics deserve the most attention when designing an IVA.

Rather than simply designing the characteristics of the agent, in this chapter we show that using facial expression and body language data from one-minute segments of a human-humman conversation (a.k.a. *thin slices*), a machine learning classifier can be trained to predict whether a novel person will experience bonding up to twenty minutes later. While it has been shown that humans have the ability to predict similar outcomes from such thin slices of an interaction (Ambady and Rosenthal, 1992), training computer algorithms to predict bonding using this type of data is a novel contribution. Data is collected unobtrusively using cameras and Microsoft Kinects while participants engage in free-form conversations. Bonding is assessed empirically using a measure adapted from the Working Alliance Inventory (Horvath and Greenberg, 1989); we show that it is strongly related to conversation quality and rapport.

To provide insight into the data we have collected and the features extracted, we provide both a qualitative and quantitative analysis of facial expression and skeletal joint position features related to bonding. We also suggest ways that an IVA could learn to synthesize the appropriate non-verbal responses based on interaction context, and provide insight into the type of non-verbal behaviors that may arise in situations in which a person is either extremely frustrated with an interaction, or deeply engaged.

6.2 *Related work*

A body of work has shown that using only thin slices (less than five minutes) of video of a person's non-verbal cues, human judges can predict everything from therapy outcomes to job performance (Ambady and Rosenthal, 1992). Since computer algorithms have successfully predicted conversational outcomes like stress and engagement using audio data (Pentland, 2004), it is possible that an IVA could use thin slices of facial expressions and body language to predict whether it is bonding with its user.

Non-verbal cues such as facial expressions and body language are a rich source of information about a person's mental state, and as such, there has been a great deal of research on how to detect, interpret, and display them. Although a thorough survey of all such work is impossible here, we refer the interested reader to a recent meta-analysis of the state of the art in automatic facial-expression recognition (Valstar et al., 2012). Automatic analysis of body-language has also been explored. For example, Avola et al. (2013) developed a system that uses Kinect data to compute features of gestural strokes, and Yang and colleagues used motion capture data to show that friendly conversational dyads had a higher degree of correlation in body language gestures (Yang et al., 2014).

Most relevant to our work is research on bonding and rapport, which has been investigated in the context of the contingency (e.g. Gratch et al. (2007)) or mirroring (e.g. Kahl and Kopp (2015)) between the VA's behavior and the user. Detailed models of rapport have also been developed (Zhao et al., 2014). Other research has investigated which facial expressions generated by an agent led to the most rapport with its users (Wong and McGee, 2012).

Personality has been examined in the context of users' reactions to an IVA (Astrid et al., 2010), and in terms of how personality similarity affects conversation quality (Cuperman and Ickes, 2009). In some cases similarity is helpful, as when partners have a similar level of *extraversion*. However, interactions between two disagreeable participants were rated as the least pleasant. While this study provides valuable insights, participants were all college students, and it is uncertain how far these claims can generalize. Our study builds upon this previous work by collecting data from participants from a diverse range of ages, ethnicities, and backgrounds, and relating personality and conversation quality to a robust measure of bonding.

6.3 *User study*

Data were collected from a study in which participants conversed while being recorded with cameras, microphones, and Microsoft Kinects; the experimental set-up is shown in Figure 6.1. To conceal the true nature of the study and ensure participants could act naturally, participants were told the purpose of the study was to train computer algorithms to read lips. They were instructed to stay within view of the recording devices, but not to over-emphasize their lip movements[1], and to keep the conversation flowing as naturally as possible. The interaction lasted for approximately 20 minutes, after which participants completed a post-study survey, were debriefed about the study's true purpose, and were compensated with $25 in Amazon gift cards for participating. All procedures were approved by the MIT IRB. In total we had 30 participants (13 male, 17 female) divided into 15 conversation dyads, recruited through the MIT Behavioral Research Lab (BRL) from the wider Boston community. There was variety across participants in terms of age (M = 40.0, SD = 15.3), occupation, ethnicity, and socioeconomic status.

Participants completed both a pre- and post-study survey. Personality traits were collected during the pre-study survey using the Big-Five

[1] Even if some participants did speak with exaggerated lip movements, this would not affect our later analysis.

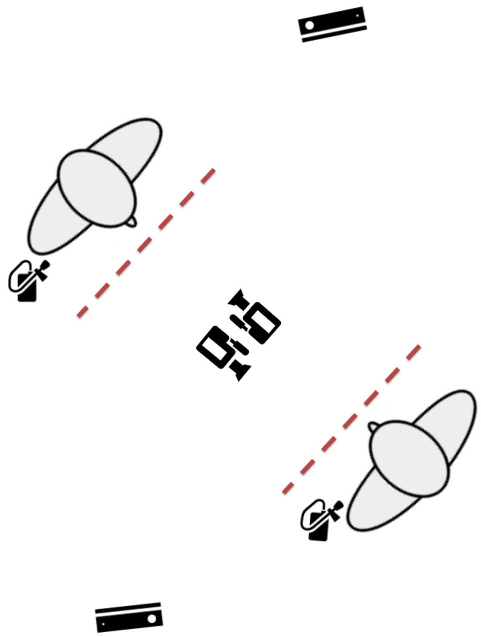


Figure 6.1: Experimental set-up for the user study, in which participants conversed while facing each other in a standing position, and were monitored with Kinects, lapel microphones, and cameras attached to a microphone stand.

Factor Markers questionnaire (Goldberg, 1992). The post-study survey contained a *Perception of Interaction* questionnaire similar to that of Cuperman and Ickes (2009), asking participants to rate their partner on a Likert scale on qualities like *interesting*, *funny*, and *attractive*. Bonding was measured with a modified version of the Bonding subscale of the Working Alliance Inventory (B-WAI). The WAI was developed to measure the degree of collaboration and trust between a therapist and client; the bonding subscale measures positive personal attachment, including “mutual trust, acceptance, and confidence" (p. 224) (Horvath and Greenberg, 1989). Items include, “My therapist and I understood each other", and “I felt uncomfortable with my therapist". The scale was adapted to our study by substituting the phrase “my partner" for “my therapist", and removing items 17, 21, and 36, which were irrelevant for short conversations between strangers. Two other items were modified slightly; Item 29 was changed to read “I had the feeling that if I said or did the wrong things, my partner would stop *talking* with me" (rather than “working with me"), and in Item 28 the phrase “my relationship” was replaced with “getting along”, such that the item reads, “Getting along with my partner was important to me”. Most items were unmodified. Typical items included “My partner and I understood each other", and “I felt uncomfortable with my partner".

6.4 *Methods*

6.4.1 *Facial expression detection*

Automated software (Affdex - Affectiva, Inc.) (McDuff et al., 2016) was applied to the videos to obtain confidence scores (from 0 to 100) indicating the presence of facial expressions. These included twelve facial action units from the Facial Action Coding System (FACS) (Ekman and Friesen, 1977), as well as smiles, lip corner pulls, seven expressions of emotion, and three axes of head pose (pitch, yaw and roll). These expressions are shown in Figure 6.2. After removing portions of the interaction in which the participant's face was not tracked, and downsampling each signal to 1 Hz to ensure smooth estimates, we obtained facial expression data for 13,714 seconds of conversation.

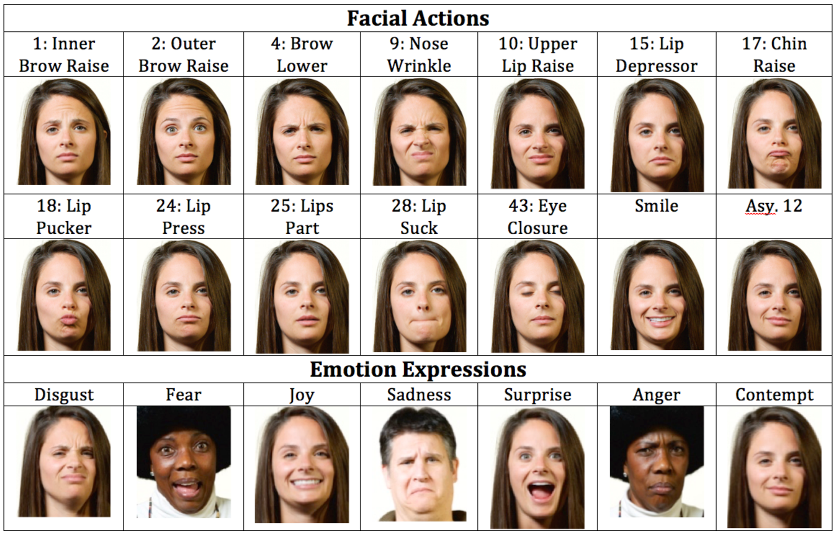


Figure 6.2: Definitions and examples of the facial actions and emotion expressions coded by Affectiva automated software (McDuff et al., 2016). The numbers correspond to the FACS codes. A similar figure can be found in McDuff (2016)

6.4.2 *Skeletal joint extraction*

Microsoft Kinects were used to gather data about the X (horizontal), Y (vertical) and Z (depth) position of participants' joints, including the head, neck, thumbs, finger tips, four positions on each limb, and three positions on the spine; the full set of joints tracked are shown in Figure 6.3. To clean this data we removed portions of the interaction in which a second body was tracked, and 4s segments in which the derivative was more than two standard deviations above the mean in any axis (X, Y or Z) (which is often due to the Kinect briefly losing track of the participant). After removing noise, minutes of the interaction that were missing more than 60% of the data were discarded, due to the unreliability of the signal during this period. The joint data was then aligned with the video data. Finally, we applied a z-score normalization to the data from each axis of each joint, which reduces effects due to the Kinect being placed in slightly different locations for different participants.

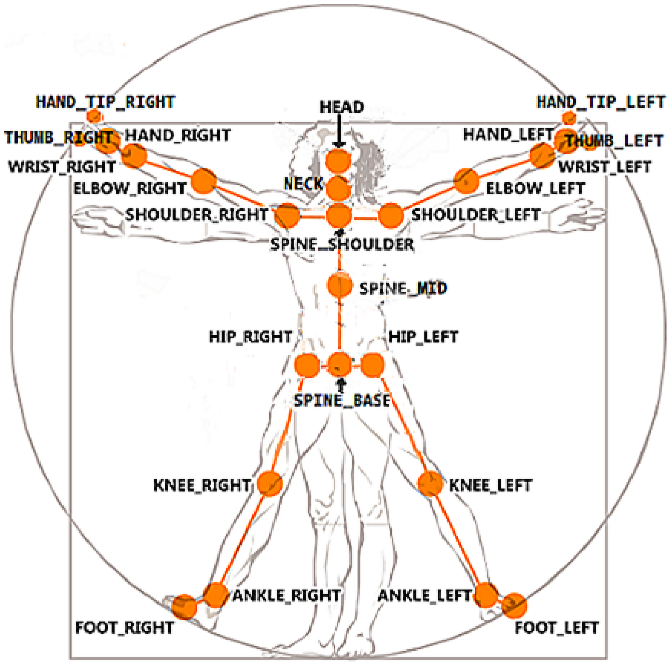


Figure 6.3: The skeletal joint positions tracked by the Microsoft Kinect. Each joint has associated X, Y, and Z coordinates.

6.4.3 *Machine learning classification*

To train our machine learning model, we extracted features from each minute of conversation for each participant and their partner. From the skeletal data, we computed five features for each joint's X, Y, and Z positions: the mean, std. dev., max. of the abs. derivative, mean derivative, and max. of the abs. second derivative. These features provide information about the position, degree of movement, speed of movement, direction of movement, and sharpness of movement (acceleration), respectively. For facial expressions, we computed the sum, mean, and std. dev. of each feature, telling us the amount, degree, and variability in expression. In total we obtained 375 joint and 143 facial expression features for each of 532 minutes of conversation.

Each minute was assigned a binary classification label, based on whether it belonged to a conversation with high or low bonding (scores were split based on the median B-WAI). The data were then randomly partitioned into training, validation, and testing sets. Data from each participant were assigned to only one set. Thus, the testing set represents completely novel, held-out data.

To reduce the number of features, we used Correlation-based Feature Selection (CFS) (Hall, 1998). CFS chooses a subset of features that are both strongly predictive of an outcome variable (in this case bonding), but also have low correlation with the rest of the features in the subset (are not redundant). CFS was applied only to the training data, to avoid contaminating the testing data. Neural network models were then trained on the CFS features using Google's TensorFlow library (Abadi et al., 2015). Both single-layer and deep architectures were explored, and parameters were tuned using the validation set.

6.5 Results

In this section we will first provide evidence establishing the reliability of the modified B-WAI, and then perform a regression analysis to determine which personality traits and characteristics are most strongly associated with bonding. We then give qualitative examples of the type of data we have collected and ways in which it can be used to detect bonding. A quantitative analysis of the differences in facial expressions and body language between participants with high and low bonding is provided. Finally, we show that machine learning can be applied to these features to accurately predict bonding levels reported by participants, up to 20 minutes later.

6.5.1 Reliability of the bonding scale

The following analysis relies on B-WAI as an aggregate measure of the bonding, rapport, and trust participants felt toward their conversational partner, as well as their feelings of warmth, comfort, and enjoyment. To examine how well B-WAI captures these characteristics, we tested the correlations between it and eight self-reported Likert-scale ratings of conversation quality (see Table 6.1). We see that B-WAI is related to participants' ratings of their partner as *interesting*, *charming*, *friendly*, and *funny*, and inversely related to their ratings of *distant* and *annoying*. After applying a Bonferroni correction, the relationships between B-WAI and *interesting*, *annoying*, and *distant* remained significant, suggesting that B-WAI is strongly related to participants' perceived conversation quality.

Measure	r	p
Interesting	**.6912**	**<.001**
Charming	.4342	.021
Friendly	.3806	.038
Funny	.3736	.046
Engaging	.1104	.561

(a) Positive correlations

Measure	r	p
Distant	**-.6207**	**<.001**
Annoying	**-.5549**	**.001**
Awkward	-.2589	.167

(b) Negative correlations

Table 6.1: Pearson's r correlations between B-WAI and conversation quality. Bolded measures are significant after performing a Bonferroni correction.

6.5.2 *Designing an agent to promote bonding*

A multiple regression analysis is employed to determine if it is possible to accurately estimate participants' B-WAI scores from information about their *partner's* personality and attitudes, and to analyze how these traits affected bonding. Although we could include factors about the participant themselves in the model, this is not under control of the designer of a virtual agent. Rather, we restrict focus to characteristics about the IVA that could be modified. Only the following traits were included: partner's Big Five extraversion and agreeableness scores, *extraversion match* (a binary variable indicating whether the pair were both introverts or both extroverts), *gender match* (defined similarly), age difference, and the participant's rating of their partner on the following qualities: attractive, funny, attentive, and excited. The resulting model statistically significantly predicted WAI score, $F(9, 19) = 4.656, p = .004$, and was able to account for 72.4% of the variance in WAI score, $R = .851$.

Table 6.2 shows the coefficients of the regression model. The first column (unstandardized β) gives the increase (or decrease) that can be expected in bonding for a 1-unit increase in the variable. For example, an increase in a participant's rating of their partner as attentive is associated with an increase of 6.024 in expected B-WAI. Three significant effects were detected; whether the gender of the two participants matched, and whether the partner was perceived as excited and attentive. It appears that bonding will be highest when the partner's gender is not a match, the partner gives the impression of listening carefully to the participant, and the partner is enthusiastic about the conversation.

Variable	**Unstd.** β	**Std. Err.**	**Std.** β	t	p
Extraversion	-4.461	3.984	-.179	-1.120	.279
Agreeableness	-5.441	6.393	-.127	-.851	.407
Extraversion match	3.158	2.169	.235	1.456	.165
Gender match	-6.765	2.923	-.393	-2.314	.034
Age difference	.150	.091	.255	1.646	.119
Attractive	.352	.788	.067	.446	.662
Funny	-1.624	1.314	-.207	-1.237	.234
Attentive	6.024	1.251	.847	4.814	.000
Excited	1.622	.754	.342	2.152	.047

Table 6.2: Linear regression coefficients for each of the factors in the model.

6.5.3 *Qualitative analysis of non-verbal cues*

In this section we provide examples of the facial expression and body language data we have collected, showing that the interaction between the two participants is highly relevant to bonding. For example, Figure 6.4 plots five minutes of facial expressions which occurred between the participant who experienced the lowest bonding in our study, P_l (top), and her partner (bottom). Although P_l began the interaction with frequent smiling, in the portion of the interaction plotted in Figure 6.4, she shows expressions of sadness as she is discussing a highly personal topic. Instead of responding empathetically, her partner continues to smile and smirk. Eventually P_l becomes angry, and afterwards simply stops emoting; for the rest of the conversation, she shows little or no facial expressions whatsoever. This interaction underlines the importance of designing an IVA to both detect emotional cues, and display the appropriate response at the right time. Further, it could suggest that an emotionally intelligent agent may need to treat a sudden suppression of affect as a potential warning sign of an upset or frustrated user.

While displaying the appropriate emotional cues in response to an unhappy user can be considered a minimum requirement of an emotionally intelligent VA, promoting a high degree of bonding and rapport can be much more subtle and complex. Figure 6.5 plots the Z position of the Spine Mid joint for the two participants in the conversation with the highest bonding. The distance maintained between the participants reveals a high degree of synchrony, suggesting they are highly attentive and responsive to each others' movements.

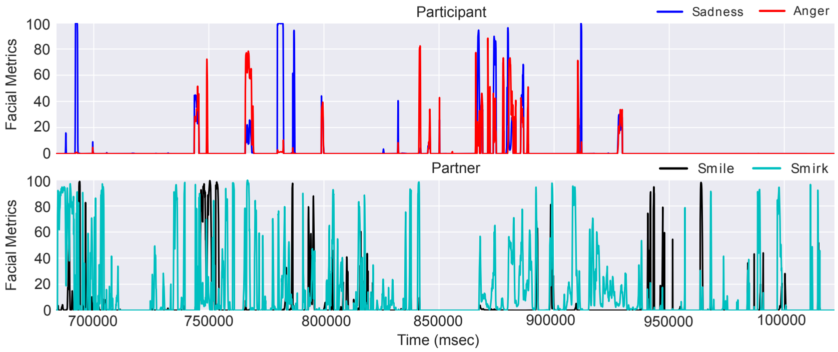


Figure 6.4: Five minutes of facial expressions from the conversation with the least bonding, in which the participant's partner fails to respond empathetically, instead smiling and smirking in response to her sadness and anger.

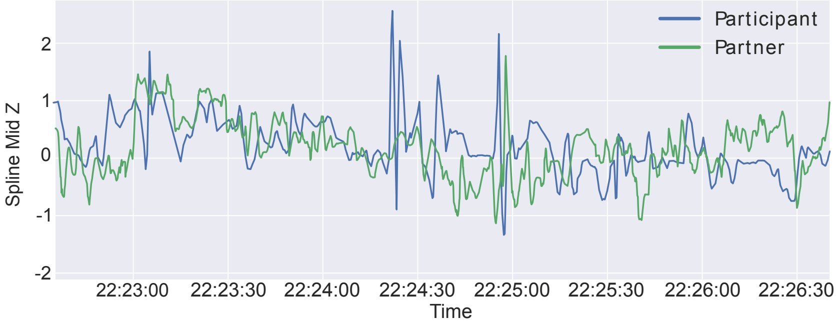


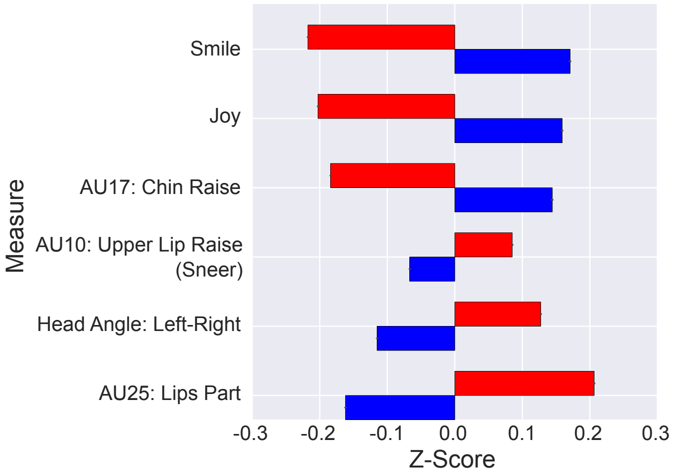
Figure 6.5: There is a strong degree of synchrony in whole-body distance (Spine Mid Z) for the pair of participants with the highest bonding.

6.5.4 *Quantitative analysis of non-verbal cues*

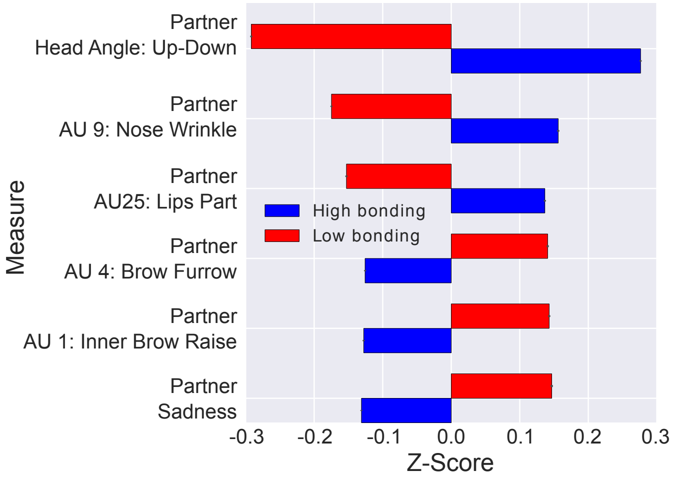
In this section we will establish what kinds of facial expression and skeletal position features are relevant to bonding, and discuss design implications for an IVA. To begin, we analyzed which facial expression behaviors are more frequent in conversations with high vs. low bonding, by computing the difference in average *z-score* between both groups. Figure 6.6 shows the three features that had the greatest difference for both high and low bonding conversations, for the participant themselves, and their partner[2]. T-tests with a Bonferroni correction were used to assess whether high and low bonding conversations differed significantly on these features, and all of them reached significance at the $\alpha = .05$ level.

Figure 6.6 reveals some expected trends. When the participant experiences bonding, she is more likely to smile, express joy, and raise her chin. When she feels that bonding is low, she is more likely to sneer and shake her head (see *Head angle left right* in Figure 6.6 (a)). In terms of the partner's behavior, frequent nodding (*Head angle up down*) and nose wrinkling is associated with higher bonding. Nose wrinkling is often detected when someone is laughing, which has

[2] The participant is the one who completes the B-WAI about their partner.



(a) Participant



(b) Partner

Figure 6.6: The facial expressions with the largest differences between conversations with high bonding (blue) and conversations with low bonding (red). If the Z-score is below zero, it means the behavior was less frequent in this group's conversations relative to the overall average. Top: participant's expression, bottom: partner's expression.

been found to be both deferential and endearing (Provine, 2001). Conversely, negative displays of emotion by the partner appear to hinder bonding; frequent brow furrows, inner eyebrow raises, or expressions of sadness are associated with lower bonding. An intriguing thing to note, however, is that bonding is not symmetric. A participant who did not enjoy an interaction could score very low on the bonding scale, even though her partner felt fine about the conversation and scored relatively high (and indeed this does occur). Therefore asymmetric effects can occur, such as with the *lips part* AU (frequently detected when a person is speaking).

Although Figure 6.6 provides some interesting insights, without accounting for interaction context it can only give an incomplete picture of facial expressions and bonding. Therefore we investigate how the contingency between conversational partners' expressions differs

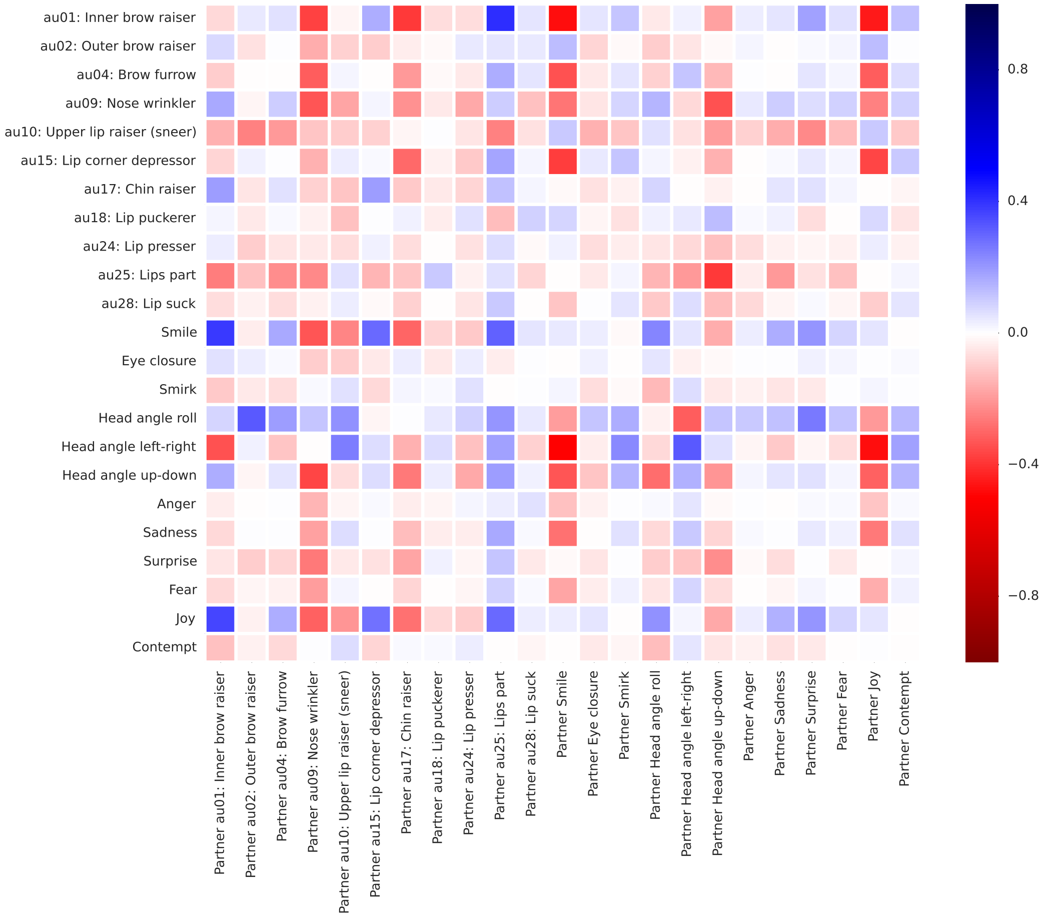


Figure 6.7: The heatmap shows the difference in correlation coefficient ($r_{diff} = r_h - r_l$) between conversations with high bonding (r_h) and low bonding (r_l). Blue tiles represent a correlation that is more strongly positive in high bonding conversations, while red represents a correlation more prevalent in low bonding.

with bonding. We computed the Pearson's r correlations between each participant's facial expressions and their partner's, for conversations with high bonding (r_h) and low bonding (r_l). The difference between these coefficients was then computed as $r_{diff} = r_h - r_l$, and plotted in Figure 6.7. Blue locations in the grid correspond to behaviors that occurred together more frequently in high bonding conversations; red locations occurred more in low bonding conversations[3]. We also tested the correlation with the partner's behavior both 1s and 5s later. The results were similar, therefore we choose to focus on the behaviors that occur together in the same second, since neural processing of facial expressions occurs on the order of 100ms (Meeren et al., 2005).

[3] Note again that bonding is not symmetric and neither is the matrix in Figure 6.7; it is computed based on the participant's perception of bonding, not her partner's.

Figure 6.7 reveals several interesting patterns. Bonding is likely to be lower if the partner is smiling or joyful while the participant is shaking her head. If the participant smiles while the partner is parting her lips the conversation is likely to have higher bonding, perhaps because the participant is enjoying what the partner is saying. Smiling at the right time appears to be important; bonding tends to be lower when the partner smiles or expresses joy in response to the participant's lip corner depressor or brow furrow. An interesting result is that there is little difference in the correlation between mutual smiling behavior in conversations with high and low bonding. This may suggest that mutual smiling is such a ubiquitous behavior that it can occur even when bonding is low. Note that there are several strong differences in inner eyebrow raising, however this AU can be associated with either sadness or happiness, making it difficult to interpret (Kohler et al., 2004).

Not only does Figure 6.7 provide insight into micro-interactions that can be used to detect bonding, it could also allow an IVA to synthesize appropriate facial responses. Consider the heatmap scores as probabilities that the agent could use when deciding what expression to display. If the user tilts her head (see the row *Head angle roll*), then the probability of the agent raising its outer eyebrows should be high, and the probability of it shaking its head should be low or almost zero. This approach is likely to be more effective than simple mirroring, because it captures the appropriateness of the expression in context.

6.5.4.1 Skeletal joint features

A similar analysis is applied to the joint data collected with the Microsoft Kinect. After performing CFS feature selection as described in Section 6.4.2, we were left with a total of 69 non-redundant[4] joint features. For each of these, we computed the *information gain*, which can be interpreted as the reduction in uncertainty about one variable obtained after observing another (Murphy, 2012). Essentially, information gain tells us which features are most predictive of bonding. The five features with the highest information gain are listed in Table 6.3.

These joint features reveal that the partner's movements in the Z direction (towards or away from the participant) are highly related to whether the participant experiences bonding. The features relate to

[4] After CFS, two body part features that are highly correlated (for example, the left and right hips) will be represented by only one of the pair (e.g. the right hip).

Feature	Info. gain	Pearson's r	p
Partner SpineBaseZ sd	0.1695	0.4190	<.001
Partner HipRightZ sd	0.1541	0.4146	<.001
Partner KneeLeftZ max abs acc	0.1219	0.3505	<.001
Partner HipRightZ max abs deriv	0.1091	0.3712	<.001
Partner HipRightZ max abs acc	0.1091	0.3712	<.001

Table 6.3: The skeletal joint features with the highest information gain. All features are significantly correlated with bonding after applying a Bonferroni correction.

the position of the partner's whole body, such as the spine, hips, and knees. Since these features describe the acceleration, variability, and speed of movement, a larger degree of movement of the partner's whole body may be more indicative of a high bonding conversation. Perhaps in conversations in which the partner is engaged and attentive, this enthusiasm is displayed by larger and more animated whole-body movements.

The synchrony between body language in conversation dyads must also be considered. As in the previous section, we computed the Pearson's r correlation between the participant's movements and her partner's in conversations with high and low bonding. Interestingly, the speed and acceleration of whole body movements are highly correlated in conversations with high bonding. Correlations in acceleration in the Z direction are in some cases quite large; for example, for the knees, $r(308) = .5226, p < .001$, hips, $r(308) = .4578, p < .001$, and spine base, $r(308) = .4465, p < .001$. This suggests that in high-bonding conversations, the partner tends to closely mirror the sharpness of the participant's movements towards or away from her. This provides supporting evidence for the hypothesis generated in the previous section, that there is a great deal of synchrony in terms of whole body movements in pairs with high bonding. Agents that can mirror whole body movements (e.g. Kahl and Kopp (2015)) may be highly effective at facilitating bonding.

6.5.5 *Predicting bonding in novel conversations.*

Using one-minute slices of the facial expression and body language features described above, we trained a series of neural network models to predict bonding, as explained in Section 6.4.3. We found that a deep architecture with 2 layers of 300 and 12 hidden nodes[5] led to the highest validation accuracy, of 64.7% (AUC=.678). Using this model, we obtained an accuracy of 85.87% and an AUC of .931 on the held-out test data, showing we can accurately predict bonding

[5] The other parameter settings were: learning rate = .01, batch size = 20, L2 regularization $\beta = .01$, no dropout.

in novel conversations. Note that 66.3% of the samples in the test set belong to high-bonding interactions, so this result is almost 20% better than the baseline majority-class classifier (always guessing the most frequent class).

While these results are promising, they should be interpreted with caution given the small size of the testing dataset (N=92, representing data from 6 participants). While the validation accuracy still exceeded the majority-class baseline of 60.78% for the validation set, it was notably lower than the test accuracy. This is likely due to the random partitioning process and the small size of the datasets. Nevertheless, because the test set comprises novel users from which the classifier has never accessed data, this serves as a proof-of-concept that it is possible for an IVA to use data collected unobtrusively from a camera and Kinect to detect whether it is bonding with a new user during each minute of the conversation. Such fine-grained sensitivity to the user's perceptions could allow an IVA to dynamically adapt to improve bonding throughout the interaction, just like an excellent human conversationalist.

6.6 *Conclusions and future work*

We have shown that facial expression and body language features can allow an IVA to detect whether or not it is bonding with its user. We also presented a matrix, learned from human high and low bonding interactions, that could allow an IVA to generate the appropriate facial expressions and body language in response to user behavior. We have shown that a machine learning classifier can be trained to predict whether a person will experience high or low bonding, given only a one-minute slice of facial expression and body language data. This information can be gathered unobtrusively with a camera and Kinect, making the classification system potentially highly useful to a future IVA.

We have compared the effects of personality, attractiveness, humour, and attitudes like excitement on bonding and rapport. We have found that bonding can be predicted effectively using personality and the traits described. Future work is needed to determine the extent to which these findings can generalize to interactions between a person and an IVA. For example, physical attraction between people could account for our finding that pairs with opposite genders have higher bonding, and these factors would presumably not be present in Human-VA interactions. However, to the extent that these findings generalize, they suggest that it may be most important to design an IVA to appear enthusiastic and attentive, rather than focusing on designing it to be agreeable, funny, attractive, or to have a similar age to the user. The importance of attentiveness may suggest that designing agents around mirroring (e.g. Kahl and Kopp (2015)) and contingent nonverbal cues (e.g. Gratch et al. (2007)) may be the most promising approaches.

As future work, the next step is to analyze the audio data, for prosody, emotional tone, and speaking turns. There are also many ways in which the modeling of the data could be improved. For example, a time-series analysis technique such as a Hidden Markov Model (Murphy, 2012) could be employed to infer the participant's mental state (bonding or not) throughout the interaction, and the joint positions could be further abstracted into higher level gestures, as described by Avola et al. (2013). Also, instead of a matrix of pair-wise measures of behavior, tensors capturing higher-order combinations could be collected from larger sets of data and used to learn and synthesize appropriate responses for outcomes beyond bonding. Even without these improvements, this work has contributed novel fundamental elements enabling the crafting of future agents with which human partners will bond.

6.7 *Statement of contributions*

To collect the conversation data in this study, I designed an experiment in consultation with Rosalind Picard, who recommended modifying the WAI and telling participants the study was about lip reading. I hired three undergraduate students to help run the study: Jenn Kim, Lisa Zahray, and Miranda McClellan. Jenn took an organizing role and ran several study sessions independently, provided valuable psychology references to help ground the study, and continued to contribute to the project after the experiment was over. The raw videos were sent to Dan McDuff to process with Affectiva to obtain facial expression time series data. I then took this data and the raw joints data and wrote code for feature extraction and the machine learning algorithms, and trained and evaluated them. The data analysis was conducted by me in consultation with Jenn and Rosalind.

7 Machine learning for working with human data

7.1 *Multimodal autoencoder* 222
7.1.1 *Introduction* 222
7.1.2 *Related work* 224
7.1.3 *Mood prediction dataset* 225
7.1.4 *Method* 227
7.1.5 *MMAE results* 232
7.1.6 *Discussion and conclusion* 236
7.2 *Automatic identification of artifacts in EDA data* 237
7.2.1 *Introduction* 237
7.2.2 *Related work* 238
7.2.3 *Methods* 240
7.2.4 *Results* 244
7.2.5 *Conclusion* 246
7.2.6 *EDA Explorer* 246
7.3 *Statement of contributions* 247

In order to learn from humans, it is necessary to collect data from them that can be used to predict their affective state, non-verbal cues, or other forms of feedback. However, collecting data from humans—especially in real-world settings—is often difficult, error-prone, noisy, and data frequently go missing. Fortunately, machine learning techniques can be used to cope with noisy or missing data. This chapter describes two settings in which we have shown that machine learning solutions can be leveraged to make human data more useful. The first project, described in Section 7.3, uses a deep learning approach to handle the scenario where a whole set of related features (a data *modality*) goes missing at once. This is relevant if, for example, you would like to make predictions about a person's mood using a model trained on data from their phone, their location, and a physiological sensor, but they forgot to wear their sensor that data.

We call this approach the *Multimodal Autoencoder*. The second project, presented in Section 7.3, deals with artifacts that can be introduced to Electrodermal Activity (EDA) data when it is collected “in the wild" via a wrist-worn sensor. We use machine learning techniques to detect such artifacts with over 95% test accuracy, which could allow a downstream system to extract clean EDA data for prediction of measures like stress and wellbeing.

7.1 Multimodal autoencoder

To accomplish forecasting of outcomes like mood in real-world situations, affective computing systems need to collect and learn from multimodal data collected over weeks or months of daily use. Such systems are likely to encounter frequent data loss, e.g. when a phone loses location access, or when a sensor is recharging. Lost data can handicap classifiers trained with all modalities present in the data. This paper describes a new technique for handling missing multimodal data using a specialized denoising autoencoder: the Multimodal Autoencoder (MMAE). Empirical results from over 200 participants and 5500 days of data demonstrate that the MMAE is able to predict the feature values from multiple missing modalities more accurately than reconstruction methods such as principal components analysis (PCA). We discuss several practical benefits of the MMAE’s encoding and show that it can provide robust mood prediction even when up to three quarters of the data sources are lost.

7.1.1 Introduction

Affective Computing studies frequently collect rich, multimodal data from a number of different sources in order to be able to model and recognize human affect. These data sources—whether they are physiological sensors, smartphone apps, eye trackers, cameras, or microphones—are often noisy or missing. Increasingly, such studies take place in natural environments over long periods of time, where the problem of missing data is exacerbated. For example, a system trying to learn how to forecast a depressed mood may need to run for many weeks or months, during which time participants are likely to not always wear their sensors, and sometimes miss filling out surveys. While research has shown that combining more data sources can lead to better predictions (D'Mello and Graesser, 2010; Kapoor and Picard, 2005), as each noisy source is added, the intersection of samples with clean data from every source becomes smaller and smaller. As the need for long-term multimodal data collection grows, especially for challenging topics such as forecasting mood, the problem of missing data sources becomes especially pronounced.

While there are a number of techniques for dealing with missing data, more often than not researchers may choose to simply discard samples that are missing one or more modalities. This can lead to a dramatic reduction in the number of samples available to train an affect recognition model, a significant problem for data-hungry machine learning models. Worse, if the data are not missing completely at random, this can bias the resulting model (Gelman and Hill, 2007).

In this paper we propose a novel method for dealing with missing multimodal data based on the idea of denoising autoencoders (Vincent et al., 2010). A denoising autoencoder is an unsupervised learning method in which a deep neural network is trained to reconstruct an input that has been corrupted by noise. In most cases, noise is injected by randomly dropping out some of the input features, or adding small Gaussian noise throughout the input vector. In contrast, we focus on the case where a whole block of features may go missing at one time – specifically, all of those features that are computed using the data from a single modality.

We demonstrate that by using a new model, which we call a Multimodal Autoencoder (MMAE), it is possible to accurately reconstruct the data from a missing modality, something that cannot be done with other techniques such as PCA. Further, we show that the MMAE can be trained with additional neural network layers designed to perform classification, effectively leveraging information from both unlabeled and labeled data. We present empirical results comparing MMAE to several other methods for dealing with missing data, and demonstrate that the MMAE consistently gives the best performance as the number of missing modalities increases.

Results are shown for the task of predicting tomorrow's mood, health, and stress, using data collected from physiological sensors, a smartphone app, and surveys. The goal of this research is to build a real-world system that can not only help participants predict their future mood and make adjustments to improve it, but also help detect early warning signs of depression, anxiety, and mental illness. However, the data inevitably contain samples with missing modalities, which can easily occur when a participant's smartphone cannot log data, or when sensor hardware malfunctions.

Previous work on this dataset (e.g. Jaques et al. (2015a,b, 2016c)) dealt with this problem by simply discarding samples for which any modality was missing. Therefore, these models cannot make accurate mood predictions if any of the data sources go missing. This is highly problematic if the models are to be used for any sort of real-world mental-health treatment and prevention program, as data frequently go missing during long-term use “in the wild".

In contrast, the new MMAE enables accurate mood prediction even with several missing modalities. Below we will show that in addition to being robust, the MMAE provides added benefits that may allow individuals with privacy or comfort concerns regarding the collection of certain types of data to opt out of providing such data, yet still enjoy the benefits of a mood forecasting system.

7.1.2 Related work

Previous research has used autoencoders to enhance emotion recognition systems. Deng and colleagues demonstrate that an autoencoder can be used to improve emotion recognition in speech through transfer learning from related domains (Deng et al., 2013). Xue and others use an autoencoder as a pre-training step in a semi-supervised learning framework to disentangle emotion from other features in speech (Xue et al., 2015). A recent, related approach uses auto-encoders for both speech emotion classification and domain adaptation, taking advantage of their ability to learn from both labeled data and unlabeled data from other domains (Deng et al., 2017).

In the medical community, denoising autoencoders have been used to effectively compress data from large, sparse, extremely noisy Electronic Health Records (EHRs) into a much smaller embedding (Miotto et al., 2016). The authors show that the autoencoder embedding can drastically improve classification accuracy over the raw and noisy feature vector, or over other dimensionality reduction techniques such as PCA.

To the best of our knowledge, no previous work has proposed using autoencoders to fill in features from missing data sources. Some

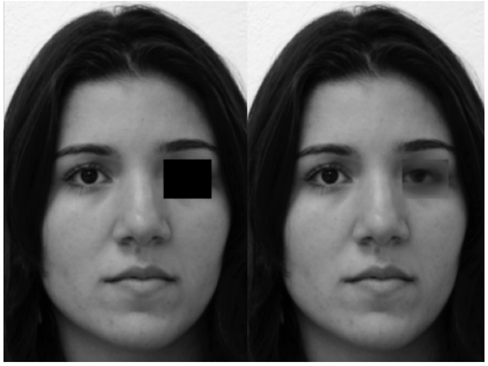


Figure 7.1: Image inpainting with an autoencoder, reproduced from Shcherbakov and Batishcheva (2014)

research that is conceptually similar to this idea comes from the computer vision community, which has investigated using autoencoders for the purpose of image inpainting (Pathak et al., 2016; Xie et al., 2012; Shcherbakov and Batishcheva, 2014). In this problem, a large swath of an image has been removed or masked, and the task of the autoencoder is to hallucinate plausible values for the missing pixels based on related images it has seen in the training data (see Figure 7.1 for an example). This task is similar to our problem, because we consider the case when many related feature values go missing at once; for example, if the smartphone app encounters an error, we can no longer compute any of the many features relating to the participant's location, calls, or SMS. However, it should be noted that image inpainting may be a considerably easier task than filling in missing sensor data, because an inpainting autoencoder can take advantage of the strong spatial regularities of images and high correlations in values of neighbouring pixels that occur in natural images, not to mention the abundance of image data that exists for unsupervised learning.

7.1.3 *Mood prediction dataset*

The task at hand is to predict individuals' mood, health, and stress tomorrow night by using today's data about their physiology and behavior. The data we use were collected as part of a large-scale study of undergraduate students entitled SNAPSHOT: Sleep, Networks, Affect, Performance, Stress, and Health using Objective Techniques (Sano, 2015). Rich, noisy, multimodal data was collected from 206 participants over 30 days each using wearable sensors, a smartphone app, and surveys. These data, along with weather information collected using DarkSky's Forecast.io API (LLC, 2016), were used to compute a total of 343 features. Only a brief overview of the data is provided here; for more details see Jaques et al. (2015b,a); Sano et al. (2015a) or Chapter 8.

Wrist-worn Affectiva Q sensors were used to collect 24-hour-a-day skin conductance (SC), skin temperature, and 3-axis accelerometer data, from which features such as step count, stillness, and SC responses (which relate to emotional arousal and stress) were extracted. Daily survey features included self-reported behaviors such as academic activities, exercise, and sleep. We include additional variables for day of the week, and whether it is a night before a school day. The smartphone app logged participants' calls, text messages, screen on/off events, and location throughout the day. In addition to extracting features about participants' communication and phone usage, location patterns were modeled with a Gaussian Mixture Model.

Each morning and evening, participants self-reported their mood (sad/happy), stress (low/high), and health (sick/healthy) on a scale from 0-100. Binary classification labels were assigned to the top and bottom 40% of these scores, discarding the middle 20% due to their questionable nature as either a 'happy' or 'sad' state, for example[1]. To predict future mood and wellbeing, features from today are combined to predict mood, stress, and health labels tomorrow night. All 5,547 days for which any data are present are divided into non-overlapping training, validation, and testing sets using a 65/20/15% split. Data from a single person may appear in multiple sets, to allow for comparison with previous work.

As with many Affective Computing studies, the multimodal, real-world nature of the dataset leads to inevitable problems with missing data, as Table 7.1 makes clear. While 206 participants × 30 days should lead to a total of 6180 days worth of data, there are only 5547 samples for which at least 40% of the features can be computed. This number is reduced significantly when we consider only those samples for which all of the multimodal data sources are available. The number of available samples drops even more precipitously when we must consider only those samples that have a supervised training label, especially when discarding the middle 20% of ratings[2].

If we wish to train a supervised learning model using only samples with all modalities, *we can use only half of the available data*. Meanwhile, valuable information contained in the remainder of the data goes to waste.

[1] Note: this is an improvement from previous work (Jaques et al., 2015a,b) in which the middle 40% of scores were discarded.

[2] The number of samples with all modalities present overlaps more heavily with labeled samples in this dataset than is typical of most datasets, since the labels are collected from a survey, and other information from this survey is considered to be one of the modalities.

Data	Num. samples	Percent
All days in the study	6180	100%
More than 40% clean data	5547	89.7%
All modalities present	3819	61.8%
Labeled	2951	47.7%
Labeled and all modalities	2886	46.7%

Table 7.1: Missing data in the SNAPSHOT study. More than half the data are lost when we require that all data sources and the classification label are present.

7.1.4 *Method*

An autoencoder is an unsupervised learning technique in which a deep neural network is trained to reproduce an input X based on the reconstruction error between X and the network's output X'; e.g. if using squared reconstruction error, the model would be trained to optimize the following loss function:

$$L(X, X') = \|X - X'\|^2 \tag{7.1}$$

A key feature of autoencoders is learning a useful representation of the data, often in a compressed format. The input $X \in \mathbb{R}^D$ must first be transformed into an *embedding* $Z \in \mathbb{R}^K$, often such that $K << D$; see Figure 7.2 for a graphical representation. The mapping from X to Z is accomplished by the *encoder* portion of the network. For example, if the encoder contains only a single neural network layer, then:

$$Z = \alpha(W_e X + b_e) \tag{7.2}$$

where W_e, b_e are the linear weights and bias and α is typically a non-linear activation function, for example a Rectified Linear Unit (ReLU).

The second half of the network, the *decoder*, maps Z to the reconstruction X'; i.e.:

$$X' = \alpha(W_d Z + b_d) \tag{7.3}$$

As a regularization technique, it is sometimes effective to tie the weights of the encoder and decoder, such that $W_d = W_e^T$.

The encoder can be considered a more complex, non-linear dimensionality reduction technique. In the simple case of a 1-layer encoder with no activation function and mean squared error (MSE) loss, the network behaves like PCA, learning to project the input in the span of the first K principle components of the data (Bengio et al., 2009).

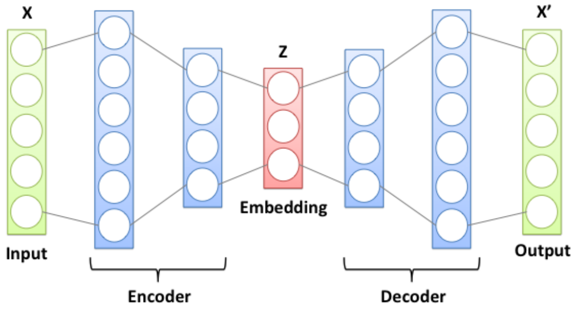


Figure 7.2: A basic autoencoder model, which learns to reconstruct X' given input X, after compressing X into the embedding representation Z.

However, with multiple hidden layers and non-linear activation functions at each layer, the embedding can learn to encode complex, higher-level features. Thus, the embedding Z can capture important conceptual information about the input data (Mikolov et al., 2013).

In a denoising autoencoder (DAE), the input X is corrupted with noise to obtain $\widetilde{X}$. The DAE is trained to reconstruct the original, noise-free input X from $\widetilde{X}$. Typically, the added noise takes the form of: a) Gaussian noise, $\widetilde{X}|X \sim \mathcal{N}(X, \sigma^2 I)$; b) masking noise, where a random fraction of the elements of X are set to 0; or c) salt and pepper noise, where a random fraction of the elements of X are set to their minimum or maximum value (Vincent et al., 2010).

7.1.4.1 MMAE

The MMAE was developed to ameliorate the likely problem where a number of contiguous features from the same modality go missing at once. We start by normalizing all of the features to be in the range $[0, 1]$. We then represent a missing modality by filling all features from that modality with the special value -1. It is important to use a special value to indicate missing data that must be filled, rather than fill with a value such as 0 which could actually occur in the real data. To train the MMAE, we first use samples that have data from every modality to provide the ground truth noise-free X. At training time, for every sample X, we compute $\widetilde{X}$ by adding noise using two methods. First, we add simple masking noise to 5% of the features, as in (Vincent et al., 2010). Second, we randomly select one or more modalities and set all of the feature values for that modality to -1; essentially, masking entire modalities at once. The model is then trained to reproduce X from $\widetilde{X}$. Effectively, this means that the model must learn to predict reasonable values for the missing modality from the rest of the features. For example, it may use the participant's physiology and location patterns to predict her survey responses, such as how much time she spent in class, or whether she drank caffeine.

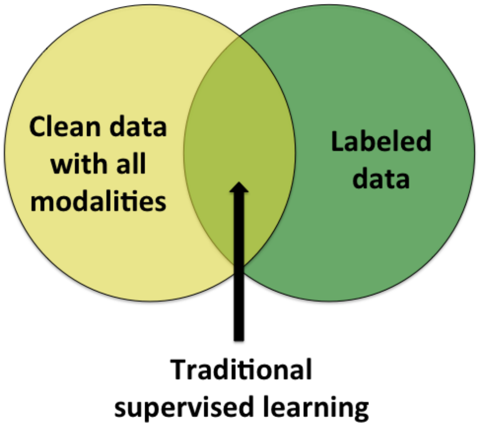

Clean data with all modalities
Labeled data
Traditional supervised learning

Figure 7.3: Data loss in traditional supervised learning paradigm

After training the autoencoder portion of the network with the clean, unsupervised examples for which all sensors are available, we then begin a second phase of training for classification. Here we connect the encoder to additional classification layers used for predicting mood, health, and stress. We allow gradients to backpropagate through the entire network, from the classification layers into the encoder. In this second phase, although we continue to add noise in the same way, we use all of the training data for which a label is available, whether it has data for every modality or not. As is presented in Figure 7.3, traditional supervised learning is only able to learn from the intersection of samples which are both clean and labeled. In contrast, the weights of the MMAE's encoder learn from both clean, unsupervised data with no labels, and noisy, supervised data with missing modalities, leveraging as much of the available data as possible.

We identify 11 modalities within the data, as in (Jaques et al., 2016c); these are shown in Table 7.2. Note that physiology is sub-divided into features from four different time intervals during the day in order to ensure each modality has a roughly similar number of features. This could allow the MMAE to more easily predict an individual's physiology in the afternoon from her physiology in the morning. However, we believe this to be a realistic scenario, since often a participant will choose not to wear the sensor for only part of a day, e.g., if s/he has to participate in an extra-curricular activity such as a dance recital or swim meet.

Still, it is possible for multiple modalities to go missing at once, e.g. all four physiology modalities. Previous research has shown that denoising autoencoders are most effective when the noise injected during training matches the actual noise in the data distribution (Xie et al., 2012). Therefore, we assessed the training data to determine how frequently each modality goes missing, and which modalities frequently go missing together. We found that in the SNAPSHOT data, the location modality is lost most frequently (likely due to participants disabling location services on their phone), and the second most likely pattern is that all of the smartphone app modalities (location, call, SMS, and screen) go missing together. We used this learned distribution over missing modalities to improve the training of the MMAE; we call this approach training with *structured noise*.

7.1.4.2 Implementation and Experiments

While using MSE is easy and most common, we found that using a cross-entropy (CE) reconstruction loss reliably led to better results for the MMAE than using MSE. The CE loss to be minimized is:

$$L_H(X, X') = -\sum_{k=1}^{D} [X_k \log X'_k + (1 - X_k) \log(1 - X'_k)]$$

Since cross-entropy is appropriate for binary values, before applying this loss we first normalized all of our features to the [0,1] range.

In addition, we experimented with implementing the MMAE as a Variational Autoencoder (VAE) (Kingma and Welling, 2013), which constrains the features in the embedding to follow K independent Gaussian distributions. This makes it more likely that a random embedding sampled from a K-dimensional multivariate Gaussian with mean 0 and variance 1, will actually correspond to a plausible sample when passed through the decoder; in other words, it makes it possible to generate new samples by interpolating in the embedding space. While this ability to generate realistic-looking samples of data is interesting, we conducted experiments using the VAE version of our MMAE and found it did not improve reconstruction or classification performance.

To assess the MMAE, we compared it to two other dimensionality reduction techniques: PCA, and a supervised feature selection technique in which the features with the highest ANOVA F-value with the classification label in the training data were selected. We

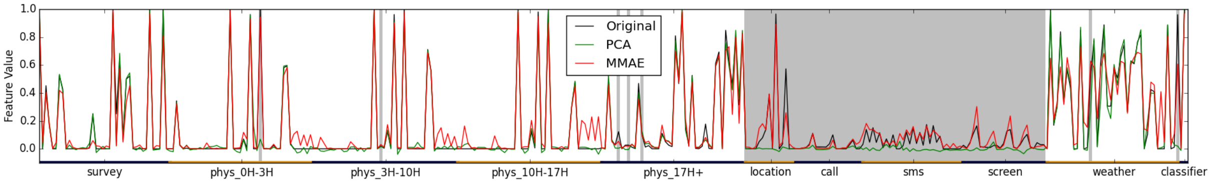


Figure 7.4: The full feature vector containing 11 modalities. MMAE reconstruction (red) and PCA reconstruction (green) are compared to the original data (black). Areas shaded grey have been masked to produce $\widetilde{X}$.

constrained each method to reduce the original 343 features to 100 dimensions to enable fair comparison; this allowed the PCA to capture 98% of the variance in the data, assuring a fair comparison. We also compared MMAE to four ways of dealing with missing data, including discarding the data when training the model, filling it with a special value like -1, filling it with the average for that feature, and filling it using a PCA reconstruction. PCA reconstruction of missing data was conducted by applying the inverse transformation learned by PCA to the 100-dimensional principle components vector.

We also compared the MMAE's classification performance to three other machine learning algorithms including Support Vector Machines (SVM), Logistic Regression (LR), and a feedforward neural network (NN). For all models we performed a grid search over possible hyperparameter settings, optimizing for performance on the validation set. Final performance on the held-out test set is reported.

The MMAE autoencoder architecture that produced the lowest reconstruction error was: hidden layers of size [300,100] for the encoder, identical structure with tied weights for the decoder, softsign activation function, no dropout, and an L2 weight regularization coefficient of .001. The MMAE architecture that produced the best classification accuracy had hidden layers of [300,100] with tied weights for the autoencoder, classification layers of [50,20], ReLU activation and dropout throughout, and an L2 weight regularization coefficient of .01 for the autoencoder, but 0 for the classification layers.

All of the Tensorflow code developed to implement the MMAE—as well as the supporting algorithms, feature selection methods, and the hyperparameter search—has been open sourced and is available at `https://github.com/natashamjaques/MultimodalAutoencoder`.

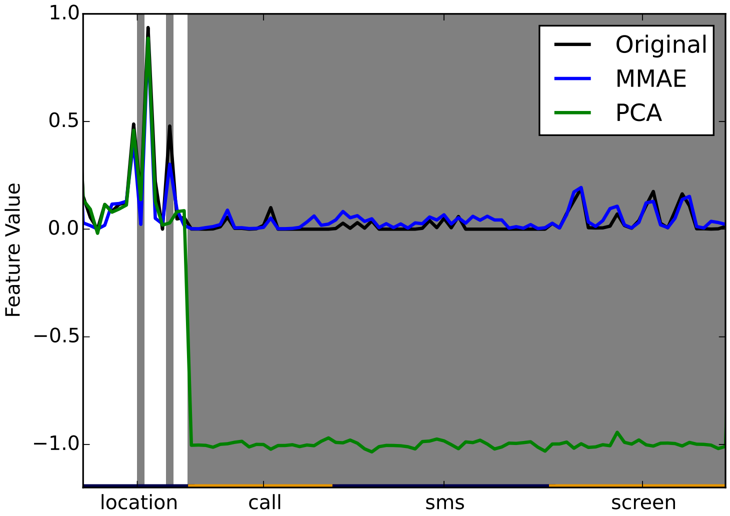


Figure 7.5: MMAE reconstruction (red), PCA reconstruction (green), original data (black). As in Figure 7.4, masked data has been shaded grey.

7.1.5 *MMAE results*

We first assess the ability of the MMAE to fill in missing modality data. As a comparison, we also reconstruct missing data with a PCA mapping learned on the training data. The PCA was able to explain 97.81% of the variance in the data when projecting down to 100 dimensions, indicating that it provides a strong baseline. In fact, PCA produces a more faithful reconstruction of the clean data than the MMAE; PCA obtains a Root Mean Squared Error (RMSE) of 0.036 on reconstructing clean test set data with no missing modalities, while MMAE scores 0.084.

However, the strength of the MMAE is its ability to restore missing modalities by predicting appropriate values based on the rest of the feature vector and similar patterns in the training data. Figure 7.4 shows that even in the case when four of the modalities go missing at the same time, the MMAE trained with structured noise is still able to predict realistic values for the features that have been masked with -1. In contrast, the PCA reconstruction hovers around whatever value was used to fill the missing data; in this case we chose a value of 0 to make it a fair comparison (since 0 is a frequent value in the real data, the RMSE will be lower if the PCA reconstruction does not differ much from the fill value), but still find that PCA is unable to reconstruct the missing features.

The difference in reconstruction performance between the MMAE and PCA is even more evident in Figure 7.5, which shows a close-up of reconstructed data from a single missing modality. PCA is able to recover one or two of the features, likely because they are highly correlated with other features in the vector which are not masked, and are thus redundant. However, in general PCA fails to reconstruct the missing data and again produces output hovering around -1 (the value used to fill missing data in this case). Conversely, the MMAE is able to accurately predict the missing feature values based on patterns learned in the training data, effectively restoring much of the original data.

7.1.5.1 Ability to reconstruct each modality

To test the MMAE's ability to reconstruct data from each of the different sources, each modality was dropped out over the entire test set, and this data was reconstructed with either an MMAE trained with by uniformly masking different modalities, or with PCA. It is clear from Table 7.2 that the MMAE produces decidedly lower RMSE when reconstructing data from a missing modality than PCA. A series of t-tests with Bonferroni correction were conducted to determine if MMAE produced significantly lower RMSE than PCA; all of the tests were significant at the $p = .001$ level.

Modality	**Num. features**	**PCA**	**MMAE**	t
Survey	39	0.363	**0.263**	26.5
Physiology 12am-3am	43	0.319	**0.095**	84.9
Physiology 3am-10am	43	0.320	**0.086**	103.7
Physiology 10am-5pm	43	0.301	**0.091**	96.4
Physiology 5pm-12am	43	0.320	**0.093**	85.3
Location	15	0.590	**0.110**	133.8
Call	20	0.280	**0.044**	137.4
SMS	30	0.481	**0.078**	154.6
Screen	25	0.423	**0.081**	149.3
Weather	40	0.488	**0.253**	82.4
Day of week, school night	2	0.634	**0.276**	12.3
Total	343	0.411	**0.134**	104.5

Table 7.2: RMSE for each modality. Bolded entries are significant improvements. We see that the MMAE significantly outperforms PCA in reconstructing missing data, reducing the total RMSE by over 75%.

From Table 7.2, it is interesting to note that the MMAE can more easily predict a person's physiology and behavioral patterns (e.g. call, sms, screen, etc.) than predict extrinsic factors like the weather or the day of the week. In particular, the RMSE for day of the week may be quite high because it is not possible to distinguish between similar week days; i.e. a student's physiology and location patterns may look the same whether it is Monday or Tuesday.

7.1.5.2 *Using the MMAE embeddings for classification*

We also tested the ability of the MMAE to produce embeddings that can be used effectively for classification. To do this each feature vector X was passed through the encoder to produce an embedding Z, then the embedding was used with other classifiers such as SVM. These results are compared to those obtained by applying other methods for dimensionality reduction or dealing with missing data; namely, PCA, feature selection, and filling the missing values with either the average or a special value like -1. Although some studies have reported dramatic improvement in prediction accuracy using autoencoder embeddings (e.g. Miotto et al. (2016)), in this case the MMAE embeddings did not approve classification performance above the comparison methods. Table 7.3 shows that the accuracy[3] in predicting mood, stress, and health on the held-out test set when using the embeddings is similar to that obtained with the other methods. A McNemar test (Adedokun and Burgess, 2011) applied revealed no significant differences. The lack of improvement is likely due to the fact that the dataset is relatively clean (only about 30% of the supervised training examples contain noise). Further, the original feature vector used in the work of Miotto and colleagues contained 100,000s of extremely noisy features (Miotto et al., 2016), whereas the 343 features from the SNAPSHOT data are already the result of several years worth of careful feature extraction, design, and selection based on domain knowledge, and are therefore already compressed and cleaned. Still, the embedding provides equivalent performance while compressing the data representation even further for enhanced computational efficiency. In addition, the embedding provided by the MMAE is a de-identified representation of otherwise highly sensitive and personal data, which may provide protection for privacy as long as the decoder is kept private.

7.1.5.3 *Robust prediction with missing modalities*

The most important use case of the MMAE is to be able to deal effectively with real-world noisy data in which several modalities may go missing at once. Therefore, we compared the MMAE to several other

[3] We also compute Area Under the Curve (AUC) scores; they are extremely similar to the accuracy scores due to the balanced nature of the classification labels.

Label	Model	Fill avg.	Fill -1	Feat. sel.	PCA	MMAE
Mood	LR	59.3	59.2	60.2	57.0	60.2
	SVM	61.8	59.5	61.2	59.3	59.1
	NN	60.3	58.2	60.9	62.5	61.5
Health	LR	59.7	59.8	57.9	56.7	58.9
	SVM	60.5	61.0	64.2	61.6	64.1
	NN	64.3	62.5	59.3	60.4	61.5
Stress	LR	62.5	61.7	59.3	59.2	60.3
	SVM	65.5	61.8	62.6	59.5	58.7
	NN	63.9	59.8	60.5	63.2	62.2

Table 7.3: Mood, health, and stress prediction accuracy on the held-out test set obtained with different approaches for filling missing data.

methods for dealing with missing data: simply discarding it and training only on clean samples, filling it with a special value like -1, or performing PCA. Each of these methods are used to train a NN, as it was shown to consistently give high performance. The MMAE can directly make predictions using the additional classification layers connected to the encoder, as described in Section 7.1.4.1.

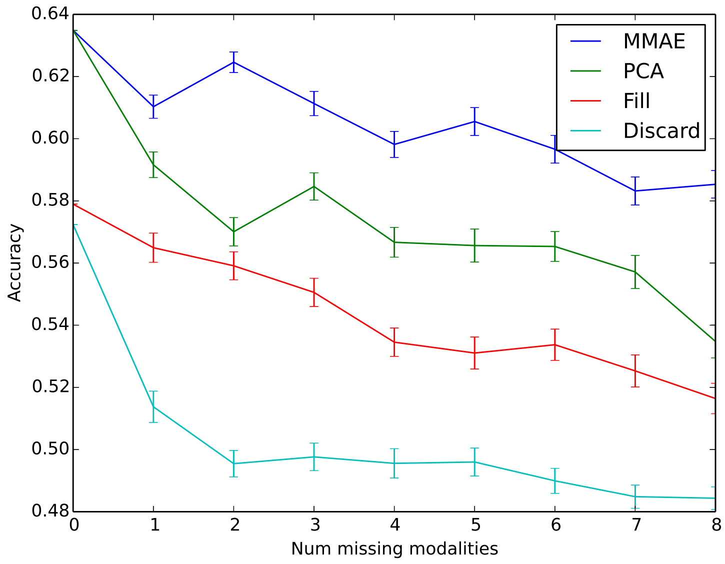


Figure 7.6: Stress prediction accuracy on the held-out test set as a function of the number of missing modalities. Error bars show 95% confidence intervals. Mood and health showed a similar pattern.

Figure 7.6 shows the performance of each of these methods on the test data as the number of modalities missing from the data increases. Note that the discard model was trained once on all available clean data, while the rest of the models were re-trained each time on training data with the appropriate number of missing modalities per row. As is obvious from Figure 7.6, the discard model—which represents the previous state-of-the-art for dealing with missing modalities in this dataset (Jaques et al., 2015a,b, 2016c)—performs extremely poorly as more modalities go missing. This is likely to reflect the performance that can be expected from such a model when applied "in the wild" in a mood prediction app. Performance is slightly higher when the NN is trained on noisy data, but in general both dimensionality reduction methods (PCA and MMAE) give higher performance from the beginning, likely because they reduce the risk of overfitting. When the dataset is relatively clean, as is the case with the SNAPSHOT data, the MMAE may not provide a significant performance improvement over PCA. However, as the number of modalities lost increases, the MMAE reliably outperforms PCA, maintaining its mood prediction accuracy even when nearly three quarters of the original features are missing. Thus, we see that the new MMAE provides an important performance advantage for a real-world system in which multiple modes of data are sporadically present.

7.1.6 Discussion and conclusion

We have described a new method for restoring missing sensor data, which is frequently lost in multimodal, real-world data collection settings. Empirical results demonstrate that the MMAE can accurately reproduce data from a lost modality, while other methods such as PCA cannot. The MMAE offers valuable new advantages for Affective Computing researchers who would like to train unbiased models on noisy data, accurately cluster noisy samples, or make robust predictions in the face of real-world data loss.

The MMAE has potential benefits in terms of providing enhanced flexibility and privacy to users of a mood prediction system. Because it can make accurate mood predictions even when data are lost, it could allow users to opt-out of providing data for all modalities. This could be particularly enticing to certain users, e.g. those who are uncomfortable wearing sensors throughout the day, or those who are concerned about privacy issues surrounding location or other data.

The MMAE also provides an effective feature reduction method that may enhance privacy; the embeddings learned by the MMAE can be used to provide roughly equal classification performance to the raw features, meaning that the raw features would not have to be stored once the embeddings are computed. The embeddings could potentially allow the highly sensitive personal data collected from this study to be shared with other researchers in a non-identifiable way.

We believe the MMAE provides an advance in the modeling of real-world mood prediction systems based on long-term multimodal data streams. Unlike prior methods, the MMAE is able to leverage valuable information from all available data, whether labeled, unlabeled, noisy, or clean. We have shown that the performance of machine learning models trained without considering missing data quickly deteriorates with data loss; however the MMAE's performance is relatively maintained even with significant loss of data. While models trained to account for missing data cannot provide reliable prediction performance as the level of noise increases, the MMAE can maintain its ability to predict tomorrow's mood even in the realistic situation where there is intermittent missing input data.

7.2 Automatic identification of artifacts in EDA data

Recently, wearable devices have allowed for long term, ambulatory measurement of electrodermal activity (EDA). Despite the fact that ambulatory recording can be noisy, and recording artifacts can easily be mistaken for a physiological response during analysis, to date there is no automatic method for detecting artifacts. This paper describes the development of a machine learning algorithm for automatically detecting EDA artifacts, and provides an empirical evaluation of classification performance. We have encoded our results into a freely available web-based tool for artifact and peak detection.

7.2.1 Introduction

Electrodermal Activity (EDA) refers to the electrical potential on the surface of the skin (Boucsein, 2012).When the body responds to stress, temperature, or exertion, the sympathetic nervous system (SNS) increases sudomotor innervation, causing EDA to increase and perspiration to occur. Because the SNS is influenced by the hypothalamus and limbic system—structures in the brain that deal with emotion—EDA has frequently been used in studies related to affective phenomena and stress (e.g. Hedman (2010); Hernandez et al.

(2011); Kappeler-Setz et al. (2013); Kocielnik et al. (2013); Reinhardt et al. (2012); Sano and Picard (2013b); Wilhelm and Roth (1998)).

Despite its popularity, little research has been done into detecting noise and artifacts in an EDA signal. This is especially problematic given the increasing number of studies that are collecting ambulatory EDA data over long time periods using wearable devices (e.g. Doberenz et al. (2011); Hedman (2010); Kappeler-Setz et al. (2013); Sano et al. (2015b); Wilhelm and Roth (1998)). While these studies may provide profound insight into how affect and stress interact with other factors in daily life, continuous and unobtrusive measurement of EDA using wearable devices makes the signal collected vulnerable to several types of noise. Artifacts can be generated from electronic noise or variation in the contact between the skin and the recording electrode caused by pressure, excessive movement, or adjustment of the device. If these artifacts remain in the signal when it is analyzed they can easily be misinterpreted and skew the analysis; for example, they may be mistaken for a skin conductance response (SCR) (a physiological reaction that may indicate increased stress).

Consequently, many researchers are forced to manually inspect the data in order to decide which portions are too noisy to retain (e.g. Fedor and Picard (2014)). This approach cannot scale to the type of large-scale EDA studies that are currently being proposed (Kappeler-Setz et al., 2013), which may involve data collected from hundreds of participants over weeks or months. In order to make collecting EDA viable in these types of studies, an automated method for detecting and removing noise and artifacts must be developed. In this project we describe the development of both a classification algorithm for automatically detecting artifacts, and an online system hosted at `eda-explorer.media.mit.edu` that will apply the algorithm to users' uploaded EDA files in order to provide them with an analysis of which portions contain artifacts.

7.2.2 *Related work*

Through extensive research into the physiological processes underlying EDA, as well as the electrical properties of the recording equipment used in measurement, Boucsein (2012) is able to provide a complete description of the characteristic shape of an SCR: the response typically lasts between 1-5 seconds, has a steep onset and an exponential decay, and reaches an amplitude of at least $.01\mu S$ (see

Fig. 7.7 for an example of a typical SCR). However, despite the availability of this knowledge, no accepted technique for removing signal artifacts has been developed.

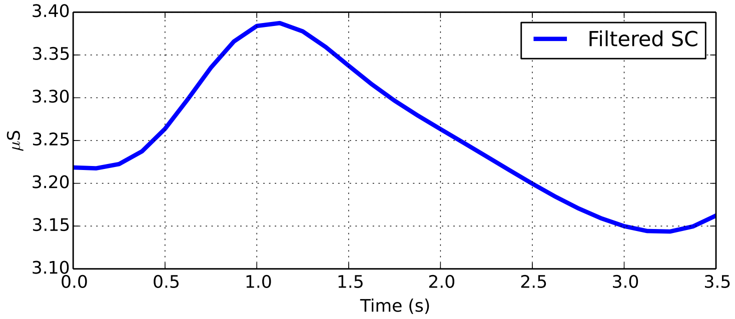


Figure 7.7: Plot of a typical Skin Conductance Response (SCR), including both shape and length.

Currently, many researchers deal with signal artifacts and noise by simply applying exponential smoothing (e.g. Hernandez et al. (2011)) or a low-pass filter (e.g. Kocielnik et al. (2013); Poh et al. (2010); Sano and Picard (2013b)). While these techniques are able to smooth small variations in the signal, they are not able to compensate for large-magnitude artifacts that can result from pressure or movement of the device during ambulatory recording. Fig. 7.8 shows a portion of signal that contains three obvious artifacts, in which the sharp decreases could not possibly be produced by human physiology. As is evident from comparing the raw and filtered versions of the signal, the low-pass filter has not removed the artifacts, and any subsequent analysis based on the filtered signal is likely to mistake the artifacts as genuine physiological responses.

Other researchers have used Boucsein's analysis to develop heuristic techniques for removing atypical portions of the EDA signal. Kocielnik and colleagues (Kocielnik et al., 2013) chose to discard portions of their data where the signal increased more than 20% per second or decreased more than 10% per second. They verified that this approach removed artifacts based on visual inspection. Using a similar approach, Storm and colleagues manually set thresholds for the maximum and minimum amplitude, maximum slope, and minimum width of an SCR, and discarded responses that did not fit these criteria (Storm et al., 2000). In another case, a study which collected EDA from two sensors (on both the ankle and the wrist) was able to detect artifacts by looking for epochs when only one of the two sensors had an abnormally low signal, or showed an unusually rapid increase or decrease (Hedman, 2010).

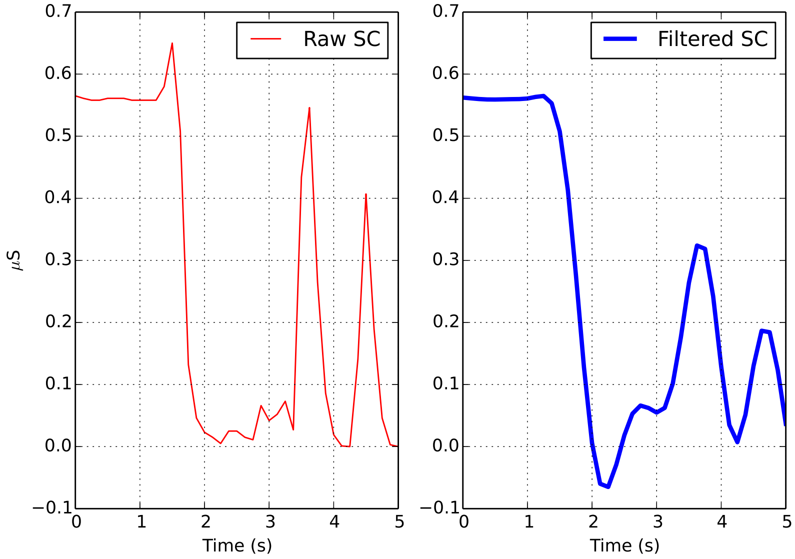


Figure 7.8: A portion of the signal containing artifacts. The raw signal is shown on the left; a 1Hz low-pass filter has been applied to the signal on the right. As is evident, the filter cannot remove the artifact.

These heuristic thresholds were developed for particular studies and participants, and verified only through visual inspection by the researchers conducting them; they may not generalize beyond those contexts. We seek to develop an empirically validated automatic technique for removing artifacts in EDA signals.

7.2.3 Methods

In order to validate our automatic artifact detection method, we needed to establish a ground truth for what portions of an EDA signal are considered clean, and what portions contain artifacts. To do this we had two expert EDA researchers label 5-second epochs of EDA data collected from a previous experiment (Fedor and Picard, 2014). The labeled data was used as input to our machine learning classifier.

7.2.3.1 Data collection

The data used in this analysis were collected during a study in which 32 participants completed physical, cognitive and emotional tasks while wearing Affectiva Q EDA sensors on both wrists (Fedor and Picard, 2014). The Q sensor collects EDA data by measuring skin conductance (SC) in microSiemens (μS) at a frequency of 8Hz. All experimental procedures were approved by the Institutional Review Board for human subjects research at MIT.

7.2.3.2 *Expert labeling*

We created a data set of 1560 non-overlapping 5-second epochs of EDA data, sampled from portions of data that were identified as possibly containing artifacts, true SCRs, or static skin conductance level (SCL). As part of our website, we built an interface to allow our two experts to review these epochs and assign a label of either 'artifact' or 'clean'. Both experts agreed on a set of criteria that defines an artifact in the signal, which is as follows:

- A peak which does not show exponential decay, depending on the context (e.g. if two SCRs occur close together in time, the first response may not decay before the second begins, yet this is not considered an artifact)
- Quantization error with $\geq 5\%$ of signal amplitude
- A sudden change in EDA correlated with motion
- A SCL ≤ 0

Although our classification labels were created using these criteria, our website provides the ability for other researchers to agree to label their own data according to their individual application needs. The site allowed the experts to view both the raw signal and a filtered signal (to which a standard 1Hz low-pass filter had been applied), as well as the accelerometer data, which is simultaneously collected by the Q sensor. We felt that viewing the accelerometer data might help the experts to identify motion artifacts. However, we do not provide acceleration data to our classification algorithm, for two reasons. Firstly, by training the classifier using only EDA data, we enable it to be applied to EDA signal collected from devices other than the Q that do not collect accelerometer data. Secondly, while it would be simple to discard portions of the signal with high power in the corresponding accelerometer data, this is not always desirable; for example, in applications such as detecting epileptic seizures, strong accelerometer signal occurs simultaneously with high EDA, but the EDA signal is both clean and valuable to the analysis (Poh, 2011).

Because we allowed the raters to skip epochs if they did not wish to label them, we eventually obtained 1301 data points that were labeled by both experts. The percentage agreement was 80.71%, and the Cohen's $\kappa = 0.55$.

There are multiple ways to deal with epochs for which the raters' labels did not agree. The first is to discard them, which is reasonable in the sense that we cannot establish a ground truth value for those epochs, meaning we have no way to train or assess the performance of the classifier. The second technique is to treat disagreements as a third class in which we are unsure whether the signal is clean or an artifact. We will present results from both approaches. Table 7.4 gives the datasets for both.

Classifier	**# Clean Epochs**	**# Questionable Epochs**	**# Artifact Epochs**
Binary	798	NA	252
Multiclass	798	251	252

Table 7.4: Number of Epochs in each classifier.

7.2.3.3 *Feature Extraction*

We extracted several features for each five second epoch. Given the importance of the shape of an SCR, we began by including statistics related to the amplitude and first and second derivative of the EDA signal (see Table 7.5). These features were computed for both the raw and filtered signal; we are not concerned about including too many features at this stage, because we later apply a feature selection procedure to reduce the chance of overfitting.

Category	**Specific Feature**
Raw SC Filtered SC	amplitude: mean 1st derivative, 2nd derivative: max, min, max of absolute value, mean absolute value
Wavelet coefficients	max, mean, standard deviation, median, number aboveZero

Table 7.5: Computed features on both the raw and filtered SC signal, as well as wavelet coefficient features.

We then used a Discrete Haar Wavelet Transform to compute additional features that may be indicative of sudden changes in the EDA signal. Wavelet Transforms have been successfully used in several noise reduction applications; because of their good time-frequency localization, they can be considered a spatially aware noise filtration technique (Xu et al., 1994). A wavelet transform decomposes a signal into coefficients at multiple scales; in our case, we obtain coefficients at 4Hz, 2Hz, and 1Hz. Because the Haar wavelet transform involves computing the degree of relatedness between subsequent points in the original signal, it is excellent for detecting edges and sharp changes (Xu et al., 1994). Using this technique applied to the participant's full EDA signal, the 3 levels of detail coefficients were computed, and statistics were computed on the coefficients over each 5-second epoch.

7.2.3.4 *Feature Selection*

Because we computed a large number of potentially redundant features, we used wrapper feature selection to ensure that our classifier did not overfit the training data. Unlike simple filtering techniques that merely rank features based on their relationship to the classification label, Wrapper feature selection (WFS) repeatedly tests subsets of features using a specific classifier[4] in order to select an independent subset of features that work well in combination with each other (Guyon and Elisseeff, 2003). Since this is computationally expensive, we used a greedy search process, which can quickly search the space of all subsets and is robust to overfitting (Guyon and Elisseeff, 2003).

7.2.3.5 *Classification*

In order to perform feature and model selection, we partitioned the data set into training, validation, and testing sets, using a randomized 60/20/20% split. Feature selection was performed using only the training data. In order to find a suitable machine learning technique for this problem, we tested a variety of algorithms including neural networks, random forests, naïve Bayes, nearest neighbour, logistic regression, and support vector machines (SVM). The algorithm that produced the best accuracy on the validation data set was SVM, so we focus on SVM for the remainder of the paper. In order to perform model selection we tested a range of settings for the parameters of SVM, including both a Radial Basis Function (RBF), polynomial, and linear kernel, and selected the settings that produced the highest accuracy on the validation set. The held-out test set was not used in feature or model selection.

[4] WFS was used with SVM after it was found to be the most effective algorithm

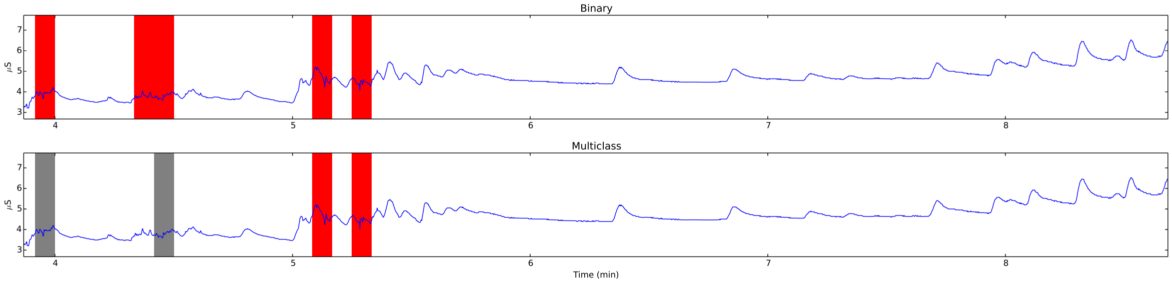


Figure 7.9: A subset of a single participant's data which includes true SCRs and artifacts. The red and grey shading shows epochs labeled as artifact and unsure, respectively. We note that both classifiers label true SCRs as clean signal.

Classifier	**Parameter settings**	**Baseline Accuracy**	**Validation Accuracy**	**Test Accuracy**
Binary	RBF, β=0.1, C=1000	76.0%	96.95%	95.67%
Multiclass	RBF, β=0.1, C=100	61.33%	88.38%	78.93%

Table 7.6: Classifier settings and accuracy on the held-out test set. The binary classifier shows very strong performance in detecting artifacts in new data.

7.2.4 Results

7.2.4.1 Classification results

Table 7.6 shows the classification results obtained for both the binary and multiclass classifiers on the validation and test sets, as well as the optimal SVM parameters. Although the accuracy for the multiclass classifier is lower (three-class classification is a more difficult problem), the output may prove more useful for real users. Fig. 7.9 shows both algorithms applied the same portion of EDA signal. As is evident from the figure, portions of the signal containing artifacts are detected (in red), while normal SCRs are labeled clean. Fig. 7.10 shows the performance of the algorithms on another sample containing a greater number of artifacts, which are also detected by both algorithms. The multiclass algorithm is able to label questionable parts of the data that are not clear artifacts in grey. Note that the binary classifier labels some epochs as artifacts that the multiclass one does not. The level of stringency needed in the classifier may depend on the researchers' application; computing aggregate measures like area under the curve may be less sensitive to artifacts than SCR detection.

7.2.4.2 Features selected

The feature selection process only led to a marginal improvement in classification on the validation set: 1.3% and 1.4% for the binary

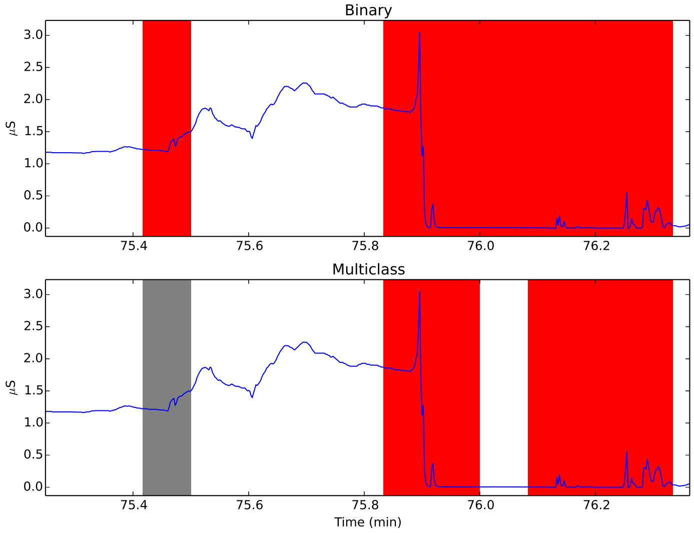


Figure 7.10: An example of a typical artifact, similar to Fig. 7.8, where the participant removed the sensor. Red and grey shading show where the classifiers labeled the SC data as artifact and questionable, respectively.

and multiclass classifiers, respectively. However the features selected provide valuable insight into the signal characteristics that best distinguish between normal EDA and an artifact. Table 7.7 shows the features selected by the binary classifier; the multiclass version selected extremely similar features. The selected features confirm the theoretical assumption that shape, including first and second derivative, are important in detecting artifacts. The wavelet features also proved valuable, especially the standard deviation of the coefficients. This is intuitive, because these values indicate whether there is a change in the wavelet domain, which may be indicative of an edge or sharp change in the original signal.

Category	Specific Feature
Raw SC	amplitude: mean 1st derivative: max absolute value 2nd derivative: max, mean absolute value
Filtered SC	amplitude: mean 2nd derivative: min, max absolute value
Wavelet	Mean: 1st coefficient St. Dev: 1st, 2nd, 3rd coefficients Median: 3rd coefficient

Table 7.7: Features Selected for Binary Classification

7.2.5 Conclusion

In summary, we have developed algorithms that can automatically and accurately distinguish artifacts in an EDA signal from normal physiological responses. The code we have written to develop these algorithms is freely available on our website, and we have built an online web-platform so that anyone can upload their raw EDA signal and receive an output indicating which portions contain noise. This tool could be enormously time-saving to researchers dealing with large data sets involving many participants measured over long periods of time. As a follow-up project, we extended our approach using active, semi-supervised learning, which can allow the machine learning algorithm to interactively ask the user to label specific epochs based on its level of uncertainty. This way, human raters will be required to label fewer epochs that are highly similar, and instead will only label novel data for which the classifier has little information.

7.2.6 EDA Explorer

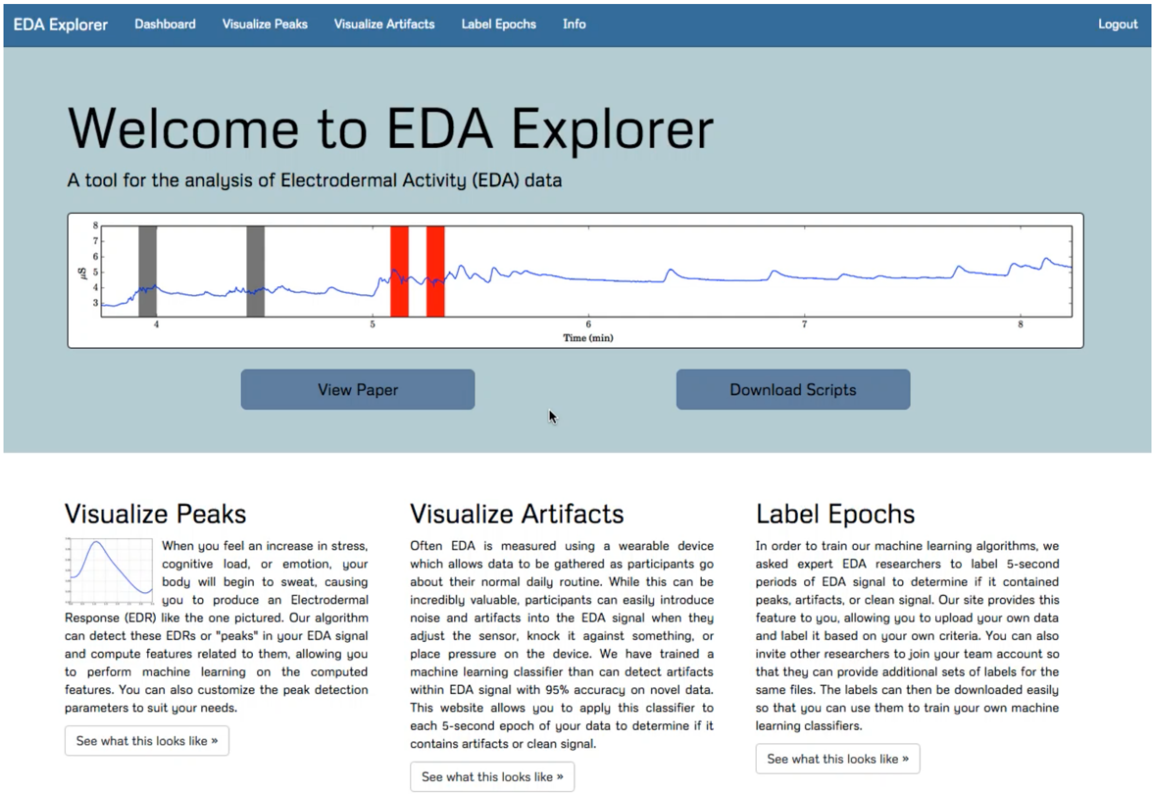
As an extension of this project, Sara Taylor and I built the web tool `eda-explorer.media.mit.edu`, which allows researchers to upload EDA data collected with various EDA sensors including the Affectiva Q and Empatica E4, and analyze it using several tools. We have maintained the site for over four and a half years now, and it currently has a total of 756 accounts, 105 of which were made this year. The below figures show how EDA Explorer allows users to apply the classification algorithms described in this section to their own data, apply our SCR-detection algorithm, and save the results. It also allows researchers to view chunks of their data and manually assign a label, which can later be used to train a classifier (this was how we obtained the expert labels for the above research).

In total, the site is currently hosting 5084 raw EDA files, 3208 artifact files, 2313 SCR files, and 334 epoch-label files. We believe EDA Explorer and the associated models and scripts have been useful in the application of EDA analysis to a variety of projects.

7.3 Statement of contributions

I developed the idea and wrote the code for the MMAE project described in Section , with advice from Rosalind Picard and Sara Taylor. Sara also contributed some code. The SNAPSHOT data used in the MMAE experiment was collected as part of a long term study involving many people, but special credit must be given to Sara Taylor, Akane Sano, and Rosalind Picard.

The project described in Section was a joint effort between myself and Sara. We began the project by meeting regularly with Akane, Szymon Fedor, Weixuan 'Vincent' Chen to devise methods for detecting artifacts in EDA data. I proposed building a website to collect expert labels, and training a machine learning classifier to predict them. Sara and I worked together to build the website, with Akane and Szymon acting as experts, using it to label data. Vincent proposed the idea of using wavelet-based features to help distinguish artifacts. Sara and I both wrote code to extract features and train machine learning algorithms, with Sara focusing more on the former and myself focusing more on the latter.



Welcome to EDA Explorer

A tool for the analysis of Electrodermal Activity (EDA) data

Visualize Peaks

When you feel an increase in stress, cognitive load, or emotion, your body will begin to sweat, causing you to produce an Electrodermal Response (EDR) like the one pictured. Our algorithm can detect these EDRs or "peaks" in your EDA signal and compute features related to them, allowing you to perform machine learning on the computed features. You can also customize the peak detection parameters to suit your needs.

See what this looks like »

Visualize Artifacts

Often EDA is measured using a wearable device which allows data to be gathered as participants go about their normal daily routine. While this can be incredibly valuable, participants can easily introduce noise and artifacts into the EDA signal when they adjust the sensor, knock it against something, or place pressure on the device. We have trained a machine learning classifier than can detect artifacts within EDA signal with 95% accuracy on novel data. This website allows you to apply this classifier to each 5-second epoch of your data to determine if it contains artifacts or clean signal.

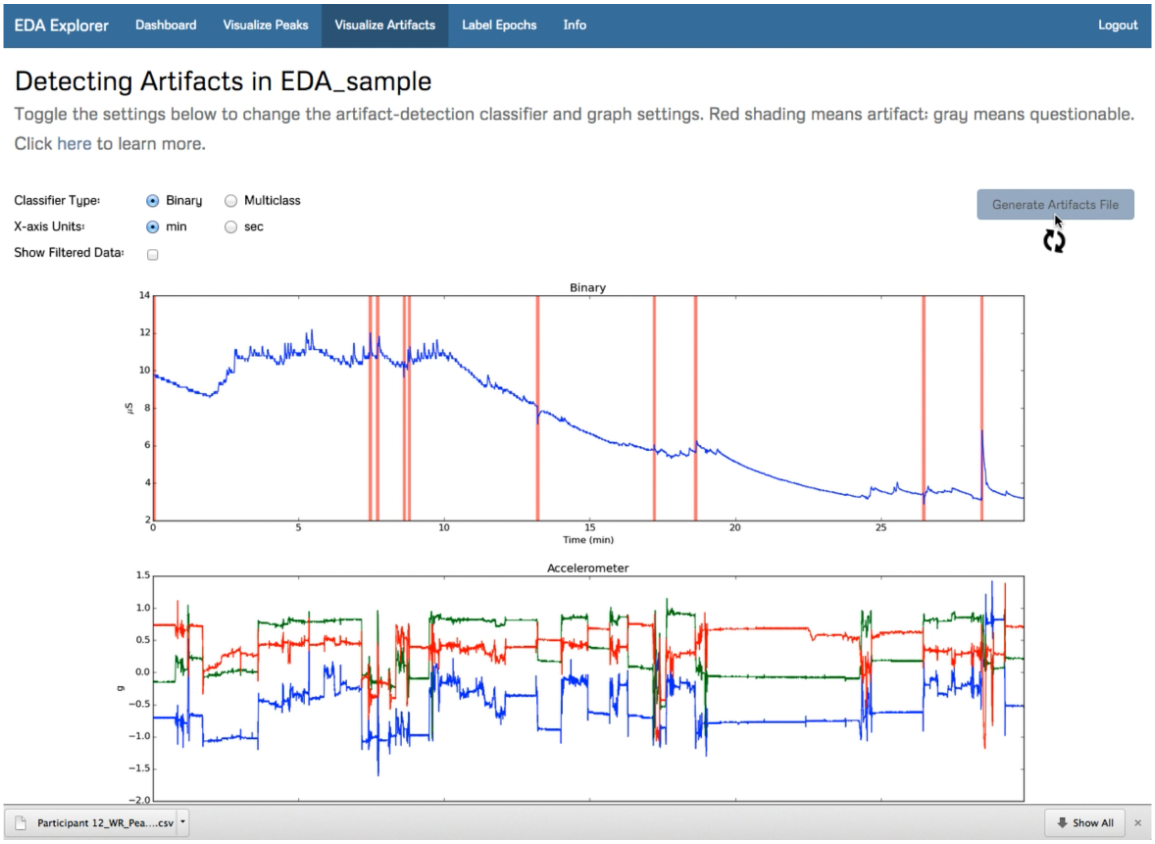
See what this looks like »

Label Epochs

In order to train our machine learning algorithms, we asked expert EDA researchers to label 5-second periods of EDA signal to determine if it contained peaks, artifacts, or clean signal. Our site provides this feature to you, allowing you to upload your own data and label it based on your own criteria. You can also invite other researchers to join your team account so that they can provide additional sets of labels for the same files. The labels can then be downloaded easily so that you can use them to train your own machine learning classifiers.

See what this looks like »

(a) Home page of EDA Explorer.

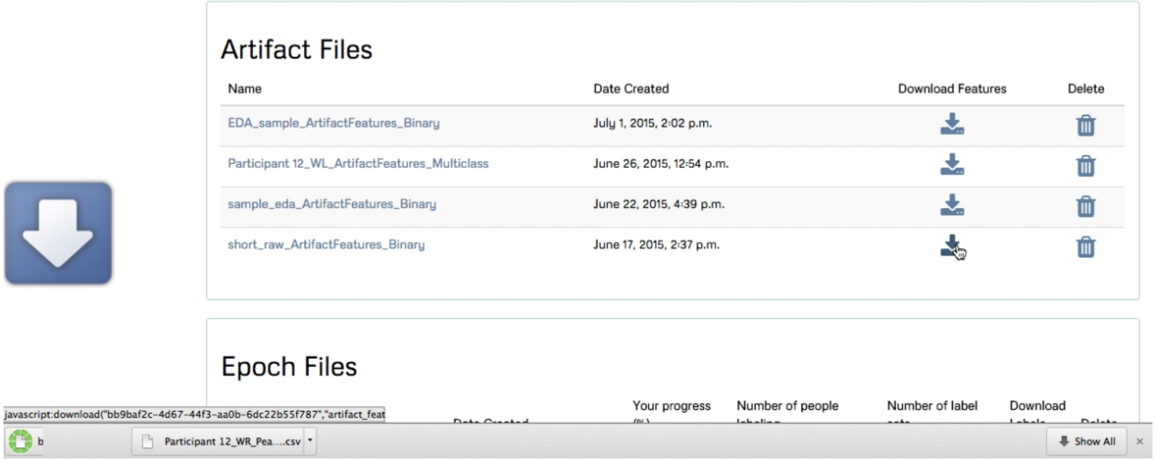


Detecting Artifacts in EDA_sample

Toggle the settings below to change the artifact-detection classifier and graph settings. Red shading means artifact; gray means questionable. Click here to learn more.

Classifier Type: ● Binary ○ Multiclass
X-axis Units: ● min ○ sec
Show Filtered Data: ☐

(b) Users can apply the trained machine learning classifiers described in this section to their own EDA data, and visualize the artifacts within the data. When ready, they can create a downloadable file marking the locations of artifacts.



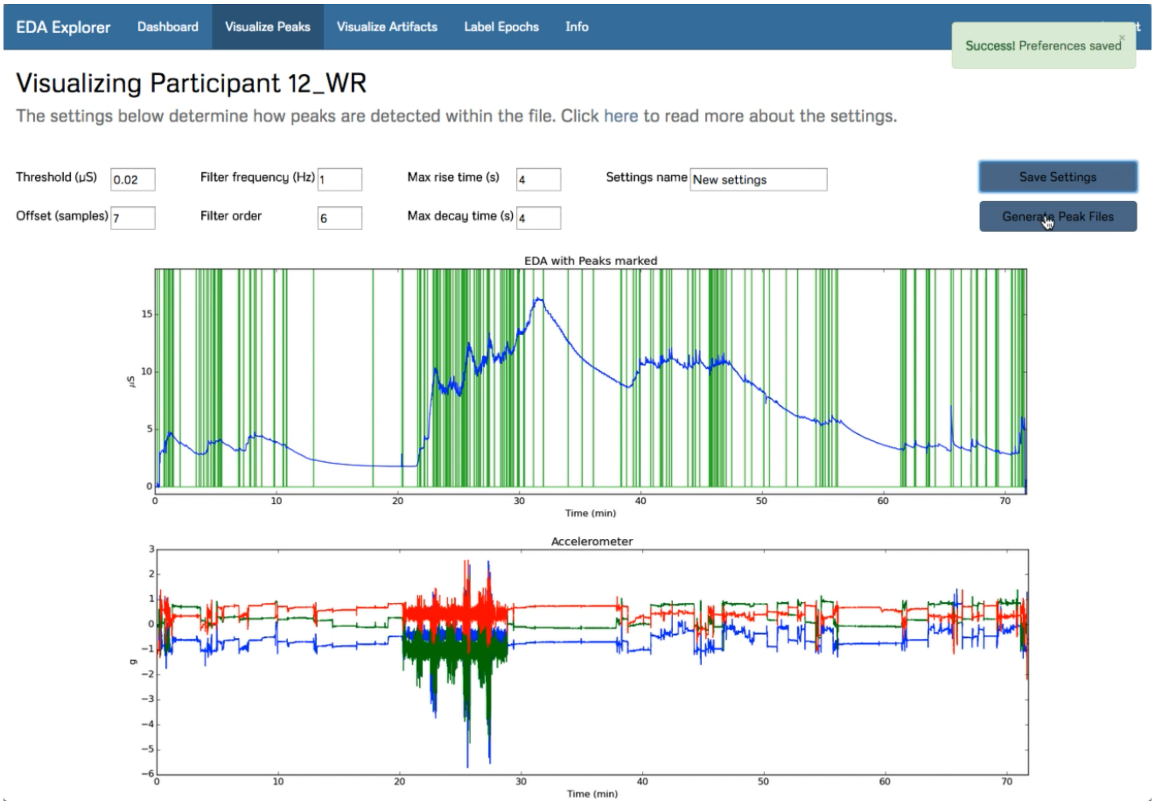
Artifact Files

Name	Date Created	Download Features	Delete
EDA_sample_ArtifactFeatures_Binary	July 1, 2015, 2:02 p.m.		
Participant 12_WL_ArtifactFeatures_Multiclass	June 26, 2015, 12:54 p.m.		
sample_eda_ArtifactFeatures_Binary	June 22, 2015, 4:39 p.m.		
short_raw_ArtifactFeatures_Binary	June 17, 2015, 2:37 p.m.		

Epoch Files

Your progress
Number of people
Number of label
Download
javascript:download("bb9baf2c-4d67-44f3-aa0b-6dc22b55f787","artifact_feat
Participant 12_WR_Pea....csv
Show All

(a) This screen shows a user downloading a saved artifact file to their computer.



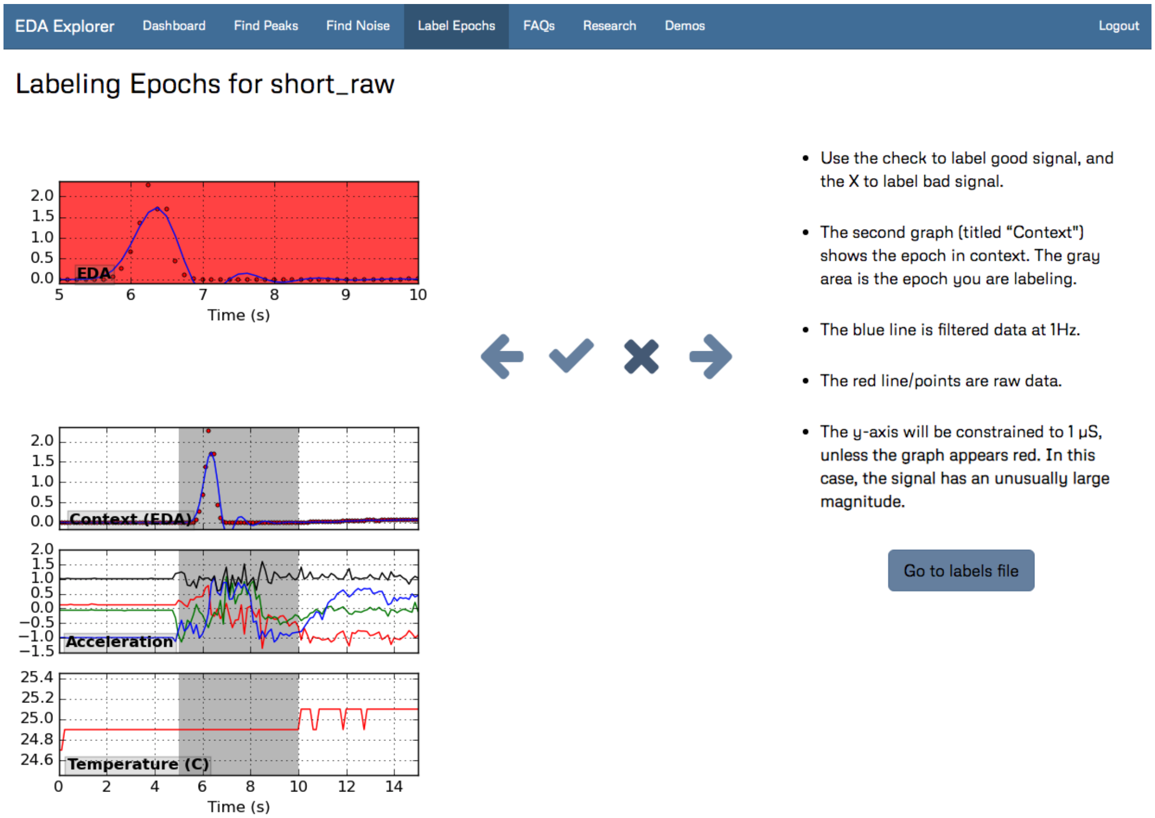
Success! Preferences saved

Visualizing Participant 12_WR

The settings below determine how peaks are detected within the file. Click here to read more about the settings.

Threshold (μS) 0.02
Filter frequency (Hz) 1
Max rise time (s) 4
Settings name New settings
Offset (samples) 7
Filter order 6
Max decay time (s) 4

(b) Users can also apply an algorithm we created to detect peaks or SCRs within their data. They can tune the hyperparameters of this algorithm to suit their needs, save these settings, and download a file of the detected peaks.



Labeling Epochs for short_raw

- Use the check to label good signal, and the X to label bad signal.
- The second graph (titled "Context") shows the epoch in context. The gray area is the epoch you are labeling.
- The blue line is filtered data at 1Hz.
- The red line/points are raw data.
- The y-axis will be constrained to 1 μS, unless the graph appears red. In this case, the signal has an unusually large magnitude.

(a) EDA Explorer also allows researchers to manually label epochs of their data, which can be used to train a machine learning classifier later. This is the process we followed to obtain expert labels for the algorithms described in this section.

8 Predicting wellbeing with personalized models

8.1 Introduction 253
8.2 Related work 257
8.2.1 Multitask Learning 258
8.2.2 Factors related to wellbeing 260
8.3 User study and feature design 261
8.3.1 Physiology 262
8.3.2 Smartphone features 263
8.3.3 Behavioral surveys and extrinsic variables 264
8.3.4 Weather 264
8.3.5 Location 264
8.3.6 Pre-study survey data 266
8.4 Initial experiments 267
8.4.1 Machine learning problem and methods 267
8.4.2 Feature evaluation 269
8.4.3 Discussion and limitations 272
8.5 Personalized multi-task learning 273
8.5.1 Neural Networks 273
8.5.2 Multi-Task Multi-Kernel Learning 275
8.5.3 Hierarchical Bayesian Logistic Regression (HBLR) 278
8.5.4 Multi-task learning experiments 281
8.5.5 Multi-task learning results and discussion 285
8.6 Personalized regression 292
8.6.1 Gaussian Processes for personalized domain adaptation 293
8.6.2 Regression experiments 296
8.6.3 Regression results 297
8.6.4 Discussion 300
8.7 Conclusions 301
8.8 Statement of contributions 303

This chapter tackles the problem of predicting next-day stress, happiness, and health using data collected from participants as they go about their daily lives, primarily via smartphones and wrist-worn sensors. There are several motivations for building a system that can accurately predict wellbeing from such unobtrusive daily monitoring. First, if a person is aware that there is an 87% chance they will feel stressed tomorrow, they can take steps to ameliorate their potential stress; for example by getting some exercise or some extra sleep. Second, any sort of AI or digital assistant that can accurately predict a person's mood is much more likely to be useful and pleasant to the user, especially if it can plan for how its actions will affect future mood, rather than just react to detecting the user's current mood. Most importantly, if a decline in happiness or an increase in stress could be predicted before it occurs, it could help to guide treatment and prevention efforts that could alleviate the risk of serious mental health consequences like depression and anxiety.

In order to model students' wellbeing, we apply machine learning methods to data collected from undergrad students monitored over the course of one month each. The data collected include physiological signals, location, smartphone logs, and survey responses to behavioral questions. Each day, participants self-reported their wellbeing in terms of stress, health, and happiness, on a scale from 0-100. Our initial experiments, which applied and compared a variety machine learning and feature selection techniques, resulted in relatively low classification accuracy ($\approx 70\%$). A review of relevant literature reveals that machine learning (ML) methods frequently yield low performance in this domain.

We posit that this is because a one-size-fits-all machine learning model is inherently ill-suited to predicting outcomes like mood and stress, which vary greatly due to individual differences. Therefore, we employ Multi-task Learning (MTL) techniques to train personalized ML models which are customized to the needs of each individual, but still leverage data from across the population. Several formulations of MTL for classification are compared: i) MTL deep neural networks, which share several hidden layers but have final layers unique to each task; ii) Multi-task Multi-Kernel learning, which feeds information across tasks through kernel weights on feature types; iii) a Hierarchical Bayesian model in which tasks share a common Dirichlet Process prior. We also extend this approach to regression models to make fine-grained predictions about an individuals' level of stress, happiness, and health, and offer the code for this work in open source. Empirical results demonstrate that using MTL to account for individual differences provides large performance improvements over traditional machine learning methods and provides personalized, actionable insights into the relationship between behavioral factors such as sleep and social activity, and individual wellbeing.

8.1 Introduction

Perceived wellbeing, as measured by self-reported health, stress, and happiness, has a number of important clinical health consequences. Stress increases susceptibility to infection and illness (Cohen et al., 1991). Self-reported health is so strongly related to actual health and all-cause mortality (Keller et al., 2012), that in a 29-year-study it was found to be the single most predictive measure of mortality, above even more objective health measures such as blood pressure readings (Aichele et al., 2016). Finally, happiness is so strongly associated with greater longevity that the effect size is comparable to that of cigarette smoking (Veenhoven, 2008).

Self-reported happiness is also indicative of scores on clinical depression measures (Cheng and Furnham, 2003). Rates of depression in the United States have notably increased in the last century, and a greater number of young adults are becoming depressed (Klerman and Weissman, 1989). Depression is prevalent on college campuses, and is also the most frequent precursor to suicide (Westefeld and Furr, 1987). Addressing depression among college students has become a major concern for some universities, especially given the fact that 18-24-year-olds have the highest incidence of suicidal ideation, and suicide has become the third leading cause of death among college-aged individuals (Kisch et al., 2005).

Clearly, the ability to model and predict subjective mood and wellbeing could be immensely beneficial, especially if such predictions could be made using data collected in an unobtrusive and privacy-sensitive way, perhaps using wearable sensors and smartphones. Such a model could open up a range of beneficial applications which passively monitor users' data and make predictions about their mental and physical wellbeing. This could not only aid in the management, treatment, and prevention of both mental illness and disease, but the predictions could be useful to any person who might want a forecast of their future mood, stress, or health in order to make adjustments to their routine to attempt to improve it. For example, if the model predicts that I will be extremely stressed tomorrow, I might want to choose a different day to agree to review that extra paper.

Unfortunately, modeling wellbeing and mood is an incredibly difficult task, and a highly accurate, robust system has yet to be developed. Historically, classification accuracies have ranged from 55-76% (e.g., Bogomolov et al. (2014); Canzian and Musolesi (2015); LiKamWa et al. (2013); Grünerbl et al. (2015)), even with sophisticated models or multi-modal data. In this chapter, we use a challenging dataset where accuracies from prior efforts to recognize wellbeing and mood ranged from 56-74% (Jaques et al., 2015a,b). Across many mood detection systems, performance remains low despite researchers' considerable efforts to develop better models and extract meaningful features from a diverse array of data sources.

We hypothesize that these models suffer from a common problem: the inability to account for individual differences. What puts one person in a good mood does not apply to everyone else. For instance, the stress reaction experienced by an introvert during a loud, crowded party might be very different for an extrovert (Brebner, 1990). Individual differences in personality can strongly affect mood and vulnerability to mental health issues such as depression (Clark et al., 1994). There are even individual differences in how people's moods are affected by the weather (Klimstra et al., 2011). The lack of ability to account for these individual differences may help to explain why many ML models tend to perform poorly in predicting mood. In fact, some authors have already found that personalization can provide important performance enhancements (e.g. Canzian and Musolesi (2015); LiKamWa et al. (2013)), although personalization typically takes the form of training many, independent models for each person. However, these approaches assume an abundance of person-specific data, which in clinical applications is not always the case. Moreover, they fail to leverage the data of all people, which can be used to build more reliable ML models.

We propose principled methods for personalizing ML models using Multi-task Learning (MTL), and show that accounting for interindividual variability via MTL can dramatically improve the prediction of these wellbeing states: mood, stress, and health. MTL is a type of transfer learning, in which models are learned simultaneously for several related tasks, but share information through similarity constraints (Caruana, 1997). We show that MTL can allow each person to have a model tailored specifically for them, which still learns from all available data. Therefore, the approach remains feasible even if there is insufficient data to train an individual machine learning model for each person. By adapting existing MTL methods to account for individual differences in the relationship between behavior and wellbeing, we are able to obtain state-of-the-art performance on the dataset under investigation (78-82% classification accuracy), significantly improving on prior published results. We extend these results to a regression setting, and show similar dramatic improvements in model fit as a result of personalization.

In addition to showing the benefits of personalization, we undertake a more challenging task than is typically attempted when modeling mood. While most prior work has focused on *detecting* current mood state, we test the ability to *predict* mood and wellbeing tomorrow night (at least 20 hours in the future), using only data from today. Specifically, assume x_t represents all the smartphone, wearable sensor, and weather data collected about a person on day t (from 12:00am to 11:59pm). Let y_t be the person's self-reported mood, stress, and health in the evening of day t (reported after 8pm). Previous work has focused on learning to model $p(y_t|x_t)$; that is, the probability of the person's current mood given the current data, which we refer to as mood *detection*. In contrast, we learn $p(y_{t+1}|x_t)$, the probability of the person's mood tomorrow given today's data. This type of prediction could be considered a type of mood *forecasting*, providing an estimate of a person's future wellbeing which could potentially allow them to better prepare for it – just as a weather forecast gives one a chance to take an umbrella rather than being left to be soaked by the rain.

Typical forecasting models make use of a history of prior labels to make predictions; i.e. such models learn the function $p(y_{t+1}|y_t, y_{t-1}, \dots, y_1, x_t, x_{t-1}, \dots, x_1)$. Using such a model for mood forecasting is less than desirable, since it implies that a person must input their mood every day in order to obtain predictions about tomorrow. In contrast, we do not use any prior labels. Instead, we learn $p(y_{t+1}|x_t)$, allowing us to predict an individual's mood without ever requiring them to manually input a mood rating.

This study also advances understanding of the role of affect in resiliency and wellbeing by investigating the relationship between factors like sleep, social and physical activity, stress, and wellbeing. We believe it is important to understand these factors, as they may contribute to resistance to depression. A body of research has shown that overall wellbeing, including factors like self-reported happiness, social support, and engagement with work, contribute to an individual's resiliency and ability to handle negative life events without becoming depressed (Seligman, 2012). Physiological variables also affect vulnerability to depression. Numerous studies have shown a significant link between sleep disturbances and subsequent depression (Tsuno et al., 2005), and physical health is strongly correlated with depression and happiness (Cheng and Furnham, 2003). Ideally we would like to investigate the factors that affect an individual's overall wellbeing both positively and negatively. Since wellbeing cannot be measured directly, we rely on self-reported measures that are known to affect wellbeing, including stress, health, and happiness. To aid our investigation we examine a wide range of data sources:

- Physiological data: electrodermal activity (EDA) (a measure of physiological stress), and 3-axis accelerometer (a measure of steps and physical activity)

- Survey data: questions related to academic activity, sleep, drug and alcohol use, and exercise

- Phone data: phone call, SMS, and usage patterns

- Location data: coordinates logged throughout the day

Finally, we propose methods for interpreting the models learned using the proposed MTL techniques, revealing how the wellbeing of different types of people responds in dramatically different ways to the same stimuli. We hope these insights can shed light on the interactions between behavior, wellbeing, and individual differences.

Our work makes the following contributions to the affective computing literature. We predict future wellbeing without requiring a history of collected wellbeing labels for each person. Our data are gathered in the “wild" as participants go about their daily lives, using surveys, wearable sensors, weather monitoring, and smartphones, and thus are relevant to use in a real-world wellbeing prediction system. We provide insights into the relationship between the collected data and mood and wellbeing. Finally, we demonstrate the ability of MTL to train personalized models that can account for individual differences, and provide the developed code for the MTL models in open source. The insight that personalization through MTL can significantly improve mood prediction performance could be a valuable step towards developing a practical, deployable mood prediction system.

8.2 Related work

The idea of estimating mood, stress, and mental health indicators using unobtrusive data collected from smartphones and wearables has been garnering increasing interest. For example, Bogomolov et al. (2013) use smartphone data combined with information about participants' personalities and the weather to detect stress with 72% accuracy. Other researchers have investigated using smartphone monitoring to detect depressive and manic states in bipolar disorder, attaining accuracy of 76% (Grünerbl et al., 2015). Detecting workplace stress is another growing body of research (Carneiro et al., 2017).

The insight that an impersonal, generic affect classifier cannot account for individual differences has been arrived at by several researchers. In estimating workplace stress, Koldijk et al. (2017) found that adding the participant ID as a feature to their model could improve accuracy in classifying mental effort. Similarly, Canzian and Musolesi (2015) found that training a generic SVM to classify depressive mood from location data and surveys resulted in sensitivity and specificity values of 0.74 and 0.78, respectively. By training an independent SVM for each person, the authors obtained values of 0.71 and 0.87.

Finally, a detailed study reported that an omnibus model trained to detect all people's mood based on smartphone communication and usage resulted in a prediction accuracy of 66% (LiKamWa et al., 2013). However, if two months of labeled data were collected for each person, then individual, independent personalized models could be trained to achieve 93% accuracy in mood classification! Since obtaining two months of training data per person can be considered somewhat unrealistic, the authors investigated methods for training a hybrid model that weights personalized examples more heavily, which can be used when there are fewer labeled training examples per person. In contrast with this work we focus on methods for making reasonable personalized predictions even in the absence of any labeled training data for a new person.

As mentioned above, almost all prior work of which we are aware has focused on mood *detection*, rather than true prediction; that is, learning $p(y_t|x_t)$, where the model label y_t and data x_t are both collected on day t. A recent paper published in April 2017 claims to be the first work to forecast future mood (Suhara et al., 2017). This work uses a Recurrent Neural Network (RNN) to predict mood given two weeks of mood history reported every day, learning the function $p(y_{t+1}|y_t, y_{t-1}...y_1, x_t, x_{t-1}...x_1)$. Using a large-scale dataset of 2,382 people, the authors achieved an AUC score of 0.886 in forecasting severely depressed mood. While a notable contribution, the drawback to this approach is that it requires a person to diligently input their mood every day. If one day is missed, a prediction cannot be made for the next two weeks. Further, the results reveal that past mood features are many times more effective at predicting future mood than any of the other data collected. Thus, using a mood history to predict future mood is a significantly easier problem. In contrast, we are able to predict tomorrow's wellbeing given a rich set of data from today ($p(y_{t+1}|x_t)$), obtaining accurate predictions about an individual's future mood through personalization, without requiring them to manually input self-reported labels.

8.2.1 Multitask Learning

MTL is a type of transfer learning, in which models are learned simultaneously for several tasks but share information through similarity constraints. Originally proposed as a way to induce efficient internal representations within neural networks (NNs) (Caruana, 1997), MTL can be used across a variety of models. It can be considered a form of regularization, and can improve generalization performance (Caruana, 1997) as long as tasks are sufficiently related (Rosenstein et al., 2005). Because MTL is beneficial when training data are scarce and noisy, it is well-suited to the messy, real-world problem of predicting mood.

Since Caruana's original work, a variety of NN MTL methods have been explored. For example, face detection accuracy for a deep convolutional network can be improved by sharing layers with networks trained on similar tasks, like face pose estimation and facial landmark localization (Zhang and Zhang, 2014). Multitasking has also been used successfully to train NNs with very little data; by using the same network to predict traffic flow in the past, present, and future, Jin and Sun were able to improve prediction accuracies using only 2112 samples of traffic data (Jin and Sun, 2008).

Hierarchical Bayesian learning is a popular approach to MTL; Baxter (1997) provide a detailed overview. The general approach is exemplified by an algorithm like *Transfer-Aware Naive Bayes* (Rosenstein et al., 2005): each task's model's parameters are drawn from a common prior distribution, thus imposing a similarity constraint. The model can update the parameters of the prior as it learns; for example, by decreasing the variance if the tasks are very similar. Bayesian inference techniques have been applied to a number of MTL scenarios. For example, MTL has been applied to a reinforcement learning problem in which each task is an environment an agent must explore, and the Markov Decision Process (MDP) learned for previous environments is treated as a strong prior on the model for a new environment (Wilson et al., 2007).

MTL has also been explored within the Affective Computing community. The idea of treating predicting the affect of a single person as a task was introduced in conjunction with Multi-Task Multi-Kernel Learning (MTMKL)(Kandemir et al., 2014) using the DEAP dataset(Koelstra et al., 2012). MTMKL is an MTL method specifically designed for Affective Computing applications which need to combine data from multiple disparate sources, or modalities (Kandemir et al., 2014). A kernel function is computed using the features from each modality, and these are combined in a weighted sum. MTL is applied by learning separate kernel weights for each task, while constraining all tasks' weights to be similar. Thus, information is shared across tasks through the kernel weights on the modalities (Kandemir et al., 2014). While treating modeling different people as related tasks in MTMKL allows for personalization, it does not allow the model to generalize to a new person. In contrast, we first cluster people based on personality and treat predicting the wellbeing of a cluster as a task, allowing us to generalize to new users who have not input wellbeing labels by placing them into the appropriate cluster.

MTL can also be applied to Affective Computing by treating outcomes like arousal and valence as the related tasks in the model. This method was used in our prior work, which applied MTMKL to the dataset under investigation in this chapter by treating the classification of happiness, stress, health, alertness, and energy as related tasks (Jaques et al., 2015b). Similarly, Xia and Liu (2015) improved the performance of a deep belief network by training it to simultaneously recognize both valence and arousal from speech. In another study of speech emotion recognition, the authors found that treating different corpora, domains, or genders as related tasks in an MTL framework offered performance benefits over learning a single model over all of the domains, or learning a separate model for each domain (Zhang et al., 2017).

8.2.2 Factors related to wellbeing

There is a growing literature showing the connection between the data collected in this study and wellbeing. Social support has been shown to mediate stress (Cohen and Wills, 1985), protect against depression (Peirce et al., 2000), and even improve overall health and recovery from illness (Cohen and Herbert, 1996). In fact, positive social relationships have been found to be the single most important factor in wellbeing in studies across ages and cultures (Reis and Gable, 2003). Conversely, people who lack social support are at risk for a range of mental health issues, including depression, anxiety, and suicide (Hawkley and Cacioppo, 2010). Because smartphone logs provide a record of the number, duration, and type of communications with social contacts, they may provide insight into an individual's social support and therefore stress, health, and happiness. Further, simply using the phone itself may affect wellbeing through sleep quality; the phone screen emits a large amount of artificial light, which has been shown to adversely affect the circadian rhythm and sleep (Czeisler et al., 1986).

In fact, smartphone data (e.g. location, proximity, and communication) have been explored in a variety of studies which are surveyed in (Lane et al., 2010). Dong et al. (2011) used smartphone data to model the underlying structure of social interactions in a student dormitory, while (Moturu et al., 2011) uses this data to explore the relationship between sleep and mood. Predicting stress from smartphone logs has been attempted by multiple researchers (e.g. Sano and Picard (2013b); Bogomolov et al. (2014); Bauer and Lukowicz (2012)). There have been preliminary studies demonstrating that mood can be classified using smartphone data (Bogomolov et al., 2014; LiKamWa et al., 2013), and Bogomolov et al. (2013) have successfully predicted happiness from a combination of smartphone data, personality, and weather patterns.

Physiological measures such as electrodermal activity (EDA) are also frequently used in studies related to affect and wellbeing (e.g. Hussain et al. (2011); Healey and Picard (1998); Arroyo et al. (2009); Vyzas (1999)). EDA measures sudomotor innervation and sweat gland activity, which is increased through activity of the sympathetic nervous system (SNS) (Poh et al., 2010). Because the SNS is influenced by the hypothalamus and the limbic system (structures in the brain that deal with emotion) EDA can be an effective technique for measuring emotion and stress. The link between EDA and stress was directly explored in a pilot version of the study used to collect the data analyzed in this chapter (Sano and Picard, 2013b). Other research has investigated the link between EDA and sleep quality (Sano and Picard, 2013a) (Sano and Picard, 2014).

8.3 *User study and feature design*

The data for this research were collected as part of a longitudinal, ambulatory study run by MIT and Brigham and Women's Hospital, investigating the impact of behavioral and physiological measures on wellbeing; more details and descriptive statistics about the data can be found in Sano et al. (2015b). The study was termed "SNAPSHOT" for its investigation of Sleep, Networks, Affect, Performance, Stress, and Health using Objective Techniques (Sano, 2015). Participants were college students who were monitored for 30 days each. The study gathers rich data, including daily smartphone, physiological, behavioral and mood data, from which we extract 343 features from the smartphone logs, location data, physiological sensor recordings, and behavioral surveys obtained about participants each day. Due to the rich, multi-scale nature of the data collected, careful feature extraction is critically important.

Following Sano and Picard (2013b), we compute each set of physiological and phone features over different time periods during the day: 12-3AM, 3-10AM, 10AM-5PM, 5-11:59PM. These intervals were determined by examining density plots of the times students were most likely to be asleep (3-10AM), or in class (10AM-5PM), as shown in Figure 8.1.

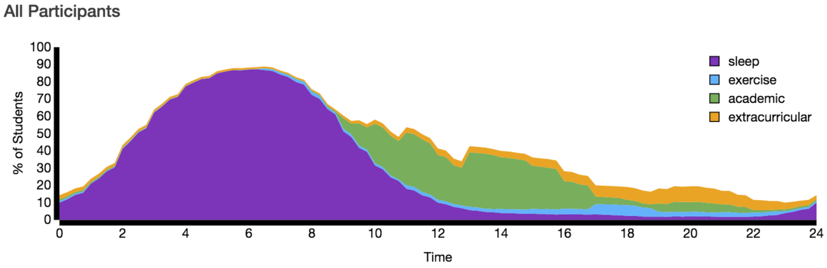


Figure 8.1: Percent of participants sleeping, studying, in extra-curricular activities, and exercising throughout the day.

8.3.1 Physiology

24-hour-a-day skin conductance (SC), skin temperature, and 3-axis acceleration were collected at 8 Hz using wrist-worn Affectiva Q sensors. SC is controlled by the sympathetic nervous system (SNS). When a person experiences a physiological stress or a "fight or flight" response, they may simultaneously experience a skin-conductance response (SCR), in which their SC signal peaks rapidly and then decays at an exponential rate (see the previous chapter for examples). Using the SC signal, we automatically remove noise using a pre-trained algorithm (as explained in the previous chapter), detect SCRs, and compute features related to their amplitude, shape, and rate, which are shown in Figure 8.2.

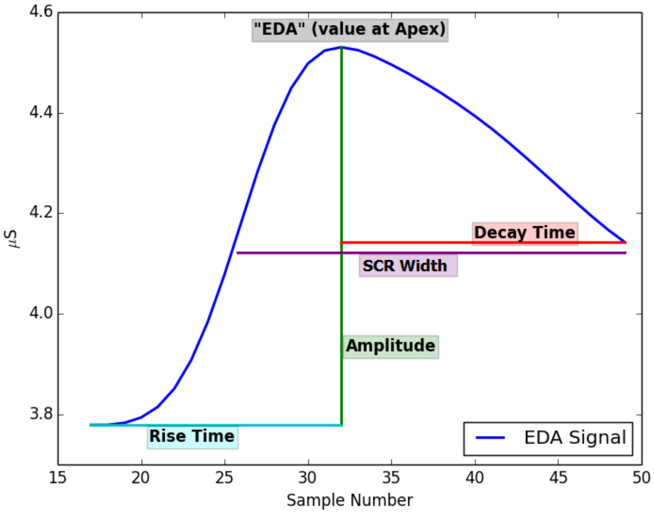


Figure 8.2: Features extracted for each non-artifact SCR detected in the SC signal.

The skin temperature and accelerometer data are also used to compute features; from the latter, we extract measures of activity, step count, and stillness. Since physical activity reduces stress and improves mood (Ratey, 2008), and skin temperature is related to the body's circadian rhythm (Partonen, 1996), we expect these features to be highly relevant. We also weight the SCR features by stillness and temperature, since we are interested in SCRs due to emotion and stress rather than exertion or heat. In total we compute 172 physiology features over different periods of the day.

8.3.2 Smartphone features

An app on participants' phones logs their calls, text messages (SMS), and whenever the phone's screen is turned on or off. Features are computed based on the timing, duration, and type of these events and the number of unique contacts with whom each person interacts. An example of SMS data is shown in Figure 8.3, in which the texting pattern on a sad day appears noticeably different than on another day.

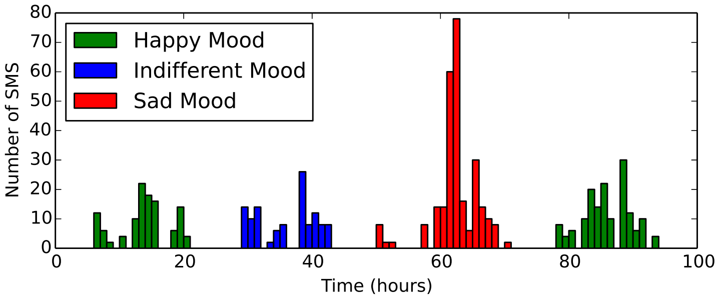


Figure 8.3: SMS frequency over four days with varying mood. The pattern of texting is markedly different on a day when the participant felt sad.

We see two mechanisms through which screen and communication information can affect wellbeing; light from the screen can disrupt circadian rhythms and therefore sleep (Czeisler et al., 1986), and the amount of social support in a person's life is strongly linked to resilience to depression (see Figure 8.3 for a possible example of a subject's social network potentially helping the subject move from a sad mood back to a happy mood.) (Seligman, 2012) (Peirce et al., 2000) (Reis and Gable, 2003). Therefore we sought to create features that would capture these factors.

We discarded days with fewer than 5 screen on events, reasoning that the app must have been malfunctioning. Previous research has shown that people interact with their phone between 10-200 times a day (Truong et al., 2014). As with physiology, the features were computed over time intervals spanning the course of the day. These may be important because we wish to capture the time when blue light from one's phone is experienced relative to the natural rhythm of sunlight.

Because we can determine whether an SMS is incoming or outgoing, we also compute the above features for these specific type of events. This could be informative, for example because incoming messages may relate more strongly to social support than outgoing messages.

Finally, we compute the number of unique callers and unique messengers for the entire day and also for each call/SMS type. Some researchers have hypothesized that diversity of social interactions with a range of individuals is linked to wellbeing (Bogomolov et al., 2014). This gives a total of 20 call, 30 SMS, and 25 screen features.

8.3.3 *Behavioral surveys and extrinsic variables*

The survey features relate to the number and duration of academic, exercise, and extracurricular activities, the amount of time spent studying, sleeping, napping, and trying to fall asleep, whether participants woke up during the night or overslept, whether they interacted with someone in person or digitally before falling asleep (referred to as *pre-sleep interaction*), and whether they had a positive or negative social interaction that day. Additionally, students indicate whether they consumed caffeine, alcohol, or drugs that could make them alert, sleepy, or tired. We are interested in how these behavioral choices and habits affect wellbeing. We include 38 features computed from the survey features, as well as 3 extrinsic variables that would be available to any smartphone app: the participant ID, the day of the week, and whether it is a school night.

8.3.4 *Weather*

Previous studies have reported on how the weather effects mood, particularily in relation to Seasonal-Affective Disorder (Partonen, 1996; Li et al., 2014). Additionally, it is well known that there are particular seasons of the year (i.e. winter) that have higher rates of poor health. Therefore, we extracted 40 features about the weather from from DarkSky's Forecast.io API (LLC, 2016). These features include information about sunlight, temperature, wind, Barometric pressure, and the difference between today's weather and the rolling average.

8.3.5 *Location*

The smartphone app logs participants' GPS coordinates throughout the day. Location is sampled whenever available in different frequencies on different devices, so we began by downsampling the signal into one set of coordinates for every 5 minute segment, computed using the median of the longitude and latitude samples within it. Segments that contained no samples were interpolated according to neighboring samples. We allowed interpolation of no more than three consequent segments (15 minutes), marking segments as missing data when necessary.

Building on previous studies (Sano and Picard, 2013b; Bogomolov et al., 2014; Bauer and Lukowicz, 2012), we extracted statistical descriptors of the subjects' distances traveled throughout the day. For each day, we computed the radius of the minimal circle enclosing the subject's location samples, as suggested by (Bogomolov et al., 2014). The source of the location data (WiFi or cellular) was used to compute an approximation of the time spent indoors and outdoors. We used the latitude and longitude coordinates of the university's campus to compute the time spent on campus each day.

Noticing that many students spent most of their time either at home or campus, we set out to model their location in a way that would better capture irregularities in this routine. We postulate that these irregularities would have a significant effect on measurements of their wellbeing. Therefore we computed a Gaussian Mixture Model (GMM) for each participants' typical location behavior. A GMM learns the number and location of Gaussian distributions required in order to collectively represent a probability distribution; in this case, the distribution over each participants' possible locations in 2-dimensional space. More formally, each participant's location distribution was modeled with K Gaussian components, as in:

$$p(x_i|\theta) = \sum_{k=1}^{K} \pi_k \mathbb{N}(x_i|\mu_k, \Sigma_k)$$

The GMM was trained on the latitude and longitude coordinates of the participant that were collected throughout the entire study (rather than just those seen on positive and negative days), since we are using the GMM to model routine behavior. The model selection process varied both the number of components, K, as well as the type of covariance matrix (spherical, diagonal, tied and full, each with different degrees of freedom). The trained model learns K Gaussian components that represent Regions of Interest (ROIs) that the participant commonly visits. We restricted the number of components K to 20, as we believed it is unlikely for an individual to have repetitive interest in more than 20 locations within a month. The best model fit was chosen using the Bayesian Information Criterion

(BIC):

$$BIC = -2\log p(D|\theta) + df(\theta)\log N$$

where θ is the MLE for the model and $df(\theta)$ is the number of degrees of freedom in the model (Murphy, 2012). Figure 8.4 illustrates a GMM fitted to the location data from one subject. We were able to verify by inspecting a map of the area that the identified components correspond to locations on the university campus and the participant's residence (specific coordinates have been redacted for privacy).

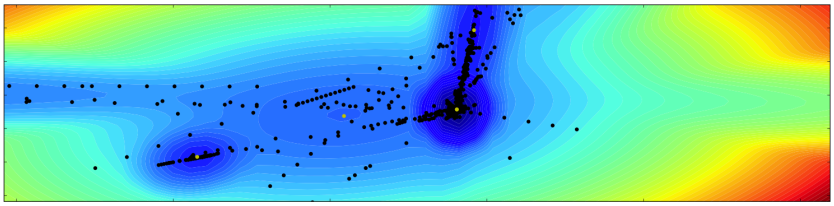


Figure 8.4: GMM fitted to location data from one participant. Black points are locations visited; the contours mark the probability distribution induced by the model, with darker blue representing more frequently visited locations.

After fitting the GMMs for each subject, they were used to compute several features that relate to the regularity of participants' routines. First, the induced probability distribution was used to compute the log likelihood for each day; this represents whether the day varied unusually from the typical routine; we refer to it as *normality of day*. Because each model learns the number and coordinates of the locations typically visited by the participants, we can determine how many different familiar locations were visited on a given day (ROIs). This approach builds on an idea that was presented by Bauer and Lukowicz (2012), where a correlation was shown between emotional stress and a person's number of geo-location ROIs. Finally, the model BIC score and Akaike Information Criterion (AIC) (Murphy, 2012) were computed using the data from each day; this represents how well the model fits that particular day, and thus how much the day deviates from routine.

8.3.6 Pre-study survey data

At the beginning of the SNAPSHOT study participants completed personality and mental health inventories. These measures included Myers-Briggs and Big Five Factor personality scores, state and trait anxiety scores, the Short-Form 12 Mental health Composite Score (MCS), Physical health Composite Score (PCS), Pittsburgh Sleep Quality Index (PSQI), the Perceived Stress Scale (PSS) and the participant's GPA and BMI (see (Sano, 2015) for details on these measures).

While these data are not incorporated directly into the models (except through the K-means clusters described in Section 8.5), we hypothesize that it may be relevant to interpreting the models learned with MTL.

8.4 Initial experiments

This section describes the initial results of applying basic machine learning classifiers to the SNAPSHOT data. The goal of this research was two-fold: 1) to understand the behavioral and physiological factors that impact wellbeing positively and negatively, and 2) to build a model that can detect when students become unhappy and thus drive interventions to mitigate the risks of depression.

8.4.1 Machine learning problem and methods

	Happiness	**Health**	**Calmness**	**Energy**	**Alertness**
Happiness	-				
Health	0.537	-			
Calmness	0.664	0.480	-		
Energy	0.480	0.410	0.389	-	
Alertness	0.374	0.318	0.324	0.721	-

Table 8.1: Correlation matrix for self-reported measures of wellbeing. All correlations are significant at the $p = 0.01$ level.

Participants in the study self-reported on five scales twice a day related to wellbeing: stress, health, energy, alertness, and happiness. Although we would ideally like to be able to predict overall wellbeing, how to create a ground-truth wellbeing measure from these scales is an open question. A first impulse might be to compute a composite measure from some of the relevant scales, for instance by computing the ratio between happiness and stress. However this schema would treat a highly happy and highly stressed state as equivalent to a low happiness and low stress state. Not only is this assumption not empirically validated, but a low happiness and low stress state could be indicative of depression (sadness and apathy), whereas a high happiness high stress state could actually represent greater underlying wellbeing than a non-stressed state, given the contribution of personal achievement and engagement with work to overall wellbeing (Seligman, 2012). For these reasons we first attempt to understand the relationship between these measures in order to frame our classification problem.

Table 8.1 shows the Pearson's correlation coefficients between all pairs of wellbeing measures. Note that stress was actually reported on a scale where a low score indicated a highly stressed state and a high score indicated calmness, so for consistency we report this as Calmness in the following explanations. We can see that all of the measures are highly related, with all correlations reaching significance at the $p = 0.01$ level, even after applying a Bonferroni correction to account for alpha inflation. Happiness has the highest correlation coefficients, suggesting that if we need to limit our predictions to on measure, Happiness will give us the most insight into the remaining scales. We are also most interested in Happiness, as it has been shown to relate directly to depression (Cheng and Furnham, 2003).

Therefore, in this initial set of experiments, we restrict our focus to classifying self-reported Happiness as our ground-truth measure. Here, we are only attempting to detect students' current happiness on the same day, rather than predict next-day happiness. Happiness was reported using a slider from "Sad" (a value of 0) to "Happy" (a value of 100). We frame the problem as binary classification; days on which a participant reported a Happiness score in the top 30% of all Happiness scores are labeled as a positive day, and days in which participants reported a Happiness score in the bottom 30% are labeled as a negative day. We do not include the middle 40% of scores. This reduces the size of our dataset to a possible 1110 points, and thus reduces our classification power. However, the behaviors on these days do not appear to have a strong effect on Happiness or wellbeing, and are thus not informative for this problem.

Using the remaining data points, we randomly partitioned a training, validation and testing dataset. The training and validation sets were used to perform feature and model selection for each data source; we refer to these sources as modalities. Given the complexity of our data, we used an iterative feature design process. After designing an initial feature set based on a review of the literature, we assessed the relevance of each feature by measuring information gain and through Wrapper Feature Selection (WFS) (Guyon and Elisseeff, 2003). Irrelevant features were removed in order to prevent overfitting, and more features were repeatedly added and assessed, until we arrived at a final feature set for each modality. The number of features eventually selected optimized accuracy on the validation set.

A variety of machine learning algorithms were tested in order to find the most appropriate model for each type of data. These include Support Vector Machines (SVM), Random Forests (RF), Neural Networks (NN), Logistic Regression (LR), k-Nearest Neighbour (kNN), Naïve Bayes, and Adaboost. After finding the best classifier, the parameter space of the classifier was searched, and the parameters which optimized performance on the validation set were selected.

8.4.2 Feature evaluation

To assess the relevance of the extracted features to self-reported happiness, we compute the information gain according to Eq. 8.1, which involves the entropy function given in Eq. 8.2.

$$\mathbb{I}(X,Y) = \mathbb{H}(X) - \mathbb{H}(X|Y) = \mathbb{H}(Y) - \mathbb{H}(Y|X) \tag{8.1}$$

$$\mathbb{H}(X) = -\sum_{i} P(x_i) \log P(x_i) \tag{8.2}$$

Information gain can be interpreted as the reduction in uncertainty about one variable after observing the other (Murphy, 2012). In this case, we assess how much information each feature provides about our classification label, Happiness. We present up to 10 of the features that had the highest information gain for each modality in Table 8.2, along with the score itself. We do not present features for which the information gain was close to zero. Since information gain is computed on each feature in isolation, it does not relate to how informative a collection of features may be when used in combination in a classification model. Therefore highly similar features (such as the multitude of screen duration features) all appear as valuable according to information gain. For this reason the information gain scores presented in Table 8.2 are not necessarily predictive of classification performance for each modality.

For interest's sake, Table 8.2 also provides an arrow indicating the direction of the relationship between the feature and the classification label, where an up arrow indicates that the feature affects happiness positively. These directions were obtained from the direction of the correlation between the feature and Happiness. For example, we see that *time indoors* is negatively correlated with Happiness; the more time spent indoors, the less likely the participant is to report feeling happy. We seek only to provide a general trend to give the reader some idea of how the feature affects Happiness, and have

Physiology	Survey	Phone	Location
0.0560 ↑ SC median 12am-6am	0.0379 ↑ Pre-sleep activity	0.0602 ↓ Screen dur. med.	0.0301 ↓ Time indoors
0.0418 ↑ SC s.d. 12am-6am	0.0240 ↑ Positive interaction	0.0456 ↓ Screen dur. med. 6pm-12am	0.0293 ↓ Normality of day
0.0408 ↑ SCR AUC total 12am-6am	0.0239 ↓ Negative interaction	0.0377 ↓ Screen dur. med. 8pm-12am	
0.0390 ↑ Mag Acc. s.d. wake	0.0200 ↑ Exercise duration	0.0367 ↓ Screen dur. med. 4pm-8pm	
0.0382 ↑ Mag Acc. s.d. 6pm-12am	0.0191 ↑ Exercise (true or false)	0.0235 ↓ Screen dur. med. 8am-12pm	
0.0381 ↑ SC med. sleep	0.0190 ↑ Exercise count	0.0213 ↓ Screen dur. mean 4pm-8pm	
0.0378 ↑ Temp. weight. SC s.d. 12am-6am	0.0140 ↓ Drugs - tired	0.0210 ↓ Screen dur. med. 12pm-6pm	
0.0374 ↑ SCR AUC mean 12am-6am	0.0128 ↓ Studying duration	0.0204 ↑ Screen total num. 8am-12pm	
0.0367 ↑ SCR AUC max 12am-6am	0.0106 ↑ Drugs - alcohol	0.0185 ↑ Screen total num.	
0.0366 ↑ SC deriv. mean 12pm-6pm	0.0105 ↓ Extracurricular count	0.0178 ↓ Screen timestamp s.d. 12am-4am	

Table 8.2: Features with the highest information for predicting happiness gain for each modality

not attempted to establish the statistical signifance of all of these relationships.

Many of the physiology features relate to the SC signal and SCRs that occur between midnight and 6am, presumably when the participant is asleep. Note that different sleep stages are characterized by different SC patterns; for example, SCRs are more likely to occur during slow-wave sleep or non-REM 2 sleep (Sano and Picard, 2013a). Therefore these features may relate to sleep quality and thus to well-being. The survey features confirm our hypotheses that exercise and social interaction are strongly linked to happiness, supporting current research on the topic (e.g. Peirce et al. (2000); Reis and Gable (2003); Ratey (2008)). It may be surprising to see that alcohol use appears to boost happiness. Since alcohol consumption is reported in the evening at the same time as the happiness scores, this is likely a reflection of the current effect of alcohol or possibly social interaction, and does not relate to prolonged alcohol use over the long term. We see from the phone logs that using the phone for longer periods of time (screen duration) appears to be associated negatively with happiness, especially if it occurs late in the day. Conversely, checking the phone frequently (screen total num) has a positive association with happiness, especially in the morning. Finally, we can see from the location features that time indoors and the likelihood or normality of the day as computed by the GMM are inversely related to happiness. This implies that when a participant spends time outdoors or deviates more from their typical routine, they tend to be happier.

8.4.2.1 *Initial classification results*

Table 8.3 presents the classification results for each modality, the relative dataset and feature set sizes for context, and the classifier and parameter settings that were found to optimize validation accuracy.

					Classification Accuracy		
Modality	**Dataset Size**	**# Features**	**Classifier**	**Parameter Settings**	**Validation**	**Baseline**	**Test**
Physiology	933	426	SVM	C=100.0, RBF kernel, $\beta = .0001$	68.37%	51.79%	64.62%
Survey	1110	32	SVM	C=100.0, RBF kernel, $\beta = .01$	71.26%	50.86%	62.50%
Phone	1072	289	RF	Num trees = 40, Max depth = infinite	66.67%	51.98%	55.95%
Location	905	15	SVM	C=100.0, RBF kernel, $\beta = 1$	69.95%	53.65%	65.10%
All	768	200	SVM	C=0.1, Linear kernel	72.84%	53.94%	68.48%

Table 8.3: Initial classification results of naively applying generic machine learning models to classify happiness. Test accuracy ranges from 56-68% across modalities. Optimal classifier and parameter settings are shown for each modality.

We found that the SVM and RF classifiers tended to produce the best results on this dataset. Accuracy on the held-out test set (i.e. the proportion of samples in which the classifier's prediction matches the true label) provides an estimation of the results we can expect on novel data; therefore we can conclude that our best model would be able to identify students that are unhappy with 68.48% accuracy.

Note that the size of the dataset involving all features is reduced due to missing data, as discussed in the previous chapter. Therefore when we combine all the modalities and restrict our focus to only those days/participants for which data from each modality is available, the dataset shrinks. This could make the 'all' dataset vulnerable to overfitting; therefore we applied the same feature selection techniques and found a reduced set of 200 features to be most effective.

Ensemble classification offers an alternative approach to training a single classifier on all of the available features. Rather, the predictions from several classifiers are integrated, often in a weighted majority vote (Rokach, 2009). We built an ensemble classifier which combines the predictions of the best classifier from the best modalities by weighting their predictions according to the classifier's validation accuracy. We found that using the best three modalities produced the highest validation accuracy. The ensemble allows us to deal with missing data in a more robust way; if a modality is missing data for a given sample day, then that classifier simply abstains from the vote. Each modality is able to maintain the maximum amount of training data, while the ensemble combines data from several modalities without losing information. The best accuracy achieved by the ensemble classifier on the held-out test set was 70.17%. Table 8.4 shows the confusion matrix for the predictions made by the ensemble classifier on the held-out test set. It is slightly more likely to falsely predict that a student is sad when she is actually happy, rather than falsely predict that a student is happy when she is actually sad. This characteristic suggests the system is more sensitive to detecting sadness, which is desirable if it is to be used to detect when to intervene if a student is becoming unhappy.

		Predicted	
		Happy	**Sad**
Actual	**Happy**	77	40
	Sad	31	90

Table 8.4: Confusion matrix for ensemble classifier

8.4.3 Discussion and limitations

Although the classifiers trained on each modality were able to achieve results exceeding the baseline, performance differed across modalities. Interestingly, location offers high performance with few features; given the features found to be the most valuable for location, it would appear that whether or not a person spends time outdoors and deviates from normal routine is strongly related to whether they will feel happy on that day. Physiology also offered relatively high performance, suggesting that wearable devices which can monitor a person's physiology throughout the day may be a promising way to detect changes in happiness, especially if those devices are capable of monitoring sleep quality.

This initial experiment demonstrated that physiological, behavioral, phone and location data can all be used to model happiness, and contributed to the literature on wellbeing by examining not only which features provide the most information about happiness and how they affect it, but also by investigating the relationship between happiness and other components of wellbeing, such as health, stress, and energy. The best accuracy obtained by our models on novel data, 70.2%, may be sufficient to guide interventions intended to prevent depression, especially if these interventions are only triggered after the classifier detects a consistent pattern of unhappiness over several days or weeks. However, this is questionable; the predictions may be so inaccurate that it could contribute to potential users distrusting the results and discontinuing use of the system.

A limitation of this work is that it does not consider individual differences; for example, extracurricular activities could make some students happy or be stressful for other students. The next section examines how to overcome this drawback through the use of multi-task learning.

8.5 *Personalized multi-task learning*

In what follows we describe several techniques for using MTL to account for interindividual differences in the relationship between behavior, physiology, and resulting mood and wellbeing. Each of the models can adapt to the specific characteristics of an individual, while still sharing information across people through a) shared layers of a deep neural network (Section 8.5.1); b) a similarity constraint on each task's classifier's weights (Section 8.5.2); or c) a common prior shared across tasks (Section 8.5.3)

The most intuitive way to use MTL to customize a model for each person is to treat a single person as a single task. However, this approach may become untenable if there are few samples per person. Since it requires that each person have a unique, fully trained, task-specific model, each person would need to provide a sufficient amount of labeled data. This may constitute a disincentivizing burden on potential users. More importantly, such a model *cannot generalize to new people or make accurate predictions about new users*.

Therefore, we begin by clustering users based on their personality and gender, and treat predicting mood for a given cluster as one prediction task. In this way, we can easily make predictions for a new user without requiring them to input their mood on a daily basis; we simply use their personality and gender to assign them to the appropriate cluster. In this study, personality is computed using the Big Five trait taxonomy (John and Srivastava, 1999), via a questionnaire that takes approximately 10 minutes to complete. We apply K-means clustering to participants' Big Five scores and gender, and assess cluster quality in an unsupervised way using *silhouette score*, which is evaluated based on the intra-cluster and nearest-cluster distance for each sample (Rousseeuw, 1987). The number of clusters which produced the highest silhouette score was $K = 37$. Using a large number of clusters allows us to create fine-grained, highly customized models that make mood predictions for extremely specific types of people.

8.5.1 *Neural Networks*

To build a MTL neural network (NN), we begin with several initial hidden layers that are shared among all the tasks. These layers then connect to smaller, task-specific layers, which are unique to each cluster. Figure 8.5 shows a simplified version of this architecture. In reality, the network can have many shared and task-specific layers.

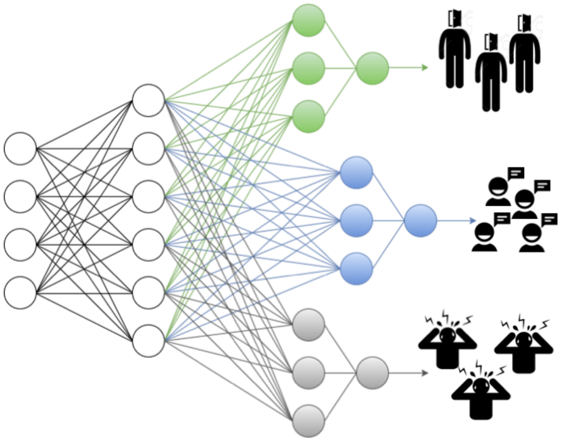


Figure 8.5: A simplified version of the MTL-NN architecture. Clusters of related people receive specialized predictions from a portion of the network trained with only their data. Shared initial layers extract features relevant to all clusters.

The intuition behind this design is that the shared layers will learn to extract information that is useful for summarizing relevant characteristics of any person's day into an efficient, generalizable embedding. The final, task-specific layers are then expected to learn how to map this embedding to a prediction customized for each cluster. For example, if the shared layers learn to condense all of the relevant smartphone app data about phone calls and texting into an aggregate measure of social support, the task-specific layers can then learn a unique weighting of this measure for each cluster. Perhaps a cluster containing participants with high extroversion scores will be more strongly affected by a lack of social support than another cluster.

To train the network, we must slightly modify the typical Stochastic Gradient Descent (SGD) algorithm. Rather than randomly selecting a mini-batch of N training samples from any of the available data, each mini-batch must only contain data from a single randomly selected task or cluster. The mini-batch is then used to predict label values y'_i, based on forward propagation through the shared weights and the appropriate cluster-specific weights. The ground-truth target labels y_i are used to compute the error with respect to that batch using the typical cross-entropy loss function:

$$L_H(\mathbf{Y}, \mathbf{Y}') = -\sum_{i=1}^{N} [y_i \log y'_i + (1 - y_i) \log(1 - y'_i)]$$

A gradient step based on the loss is then used to update both the cluster-specific weights, as well as to adjust the weights within the shared layers. By continuing to randomly sample a new cluster and update both the cluster-specific and shared weights, the network will eventually learn a shared representation relevant to all clusters.

While deep learning is a powerful branch of ML, when training on small datasets such as the one under discussion in this chapter it is important to heavily regularize the network to avoid overfitting. Although MTL itself is a strong form of regularization, we implement several other techniques to ensure generalizable predictions. As is common, we include the following L2 regularization term in the loss function: $-\beta\|\mathbf{W}\|_2^2$, where $\mathbf{W}$ are the weights of the network. We also train the network to simultaneously predict all three wellbeing labels to further improve the generalizability of the embedding. Finally, we implement dropout, a popular approach to NN regularization in which some portion of the network's weights are randomly "dropped out" (set to 0) during training. This forces the network to learn redundant representations and is statistically very powerful. Using a dropout factor of 0.5 (meaning there is a 50% chance a given weight will be dropped during training) on a NN with n nodes is equivalent to training 2^n NNs which all share parameters (Srivastava et al., 2014). This is easy to verify; consider a binary variable that represents whether or not a node is dropped out on a given training iteration. Since there are n nodes, there are 2^n possible combinations of these binary variables. Moreover, each of these sub-networks are trained on different, random mini-batches of data, and this bagging effect further enhances generalization performance.

8.5.2 *Multi-Task Multi-Kernel Learning*

As introduced in Section 8.2, the MTMKL algorithm developed by Kandemir et al. (2014) is a MTL technique designed for the problem of classifying several related emotions (tasks) based on multiple data modalities (Kandemir et al., 2014). MTMKL is a modified version of Multi-Kernel Learning (MKL) in which tasks share information through kernel weights on the modalities. Here, we consider the problem of using MTMKL to build a personalized model that can account for individual differences. Therefore, we treat each task as predicting the wellbeing for one cluster; that is, a group of highly similar people which share the same gender or personality traits.

MTMKL uses a least-squares support vector machine (LSSVM) for each task-specific model. Unlike the canonical SVM, the LSSVM uses a quadratic error on the "slack" variables instead of an L1 error.

As a result, the LSSVM can be learned by solving a series of linear equations in contrast to using quadratic programing to learn an SVM model. The LSSVM has the added benefit that when only a single label is present in the training data, its predictions will default to predict only that label.

The LSSVM can be learned by solving the following optimization problem, in which N is the total number of samples, x_i is the ith feature vector, y_i is the ith label, $k(x_i, x_j)$ is a kernel function, and α is the set of dual coefficients as in a conventional SVM:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(x_i, x_j) \\ & \qquad - \frac{1}{2C}\sum_{i=1}^{N}\alpha_i^2 + \sum_{i=1}^{N}\alpha_i \\ \text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \end{aligned}$$

In MKL, we build on the LSSVM by adjusting the kernel function. In particular, we use a kernel function to compute the similarity between feature vectors for each modality m, and the kernels are combined using a weighted sum. The weights depend on the usefulness of each modality in prediction. That is, more useful modalities will have larger kernel weights so that differences in that data modality are more helpful in prediction.

Concretely, we assign a kernel k_m to the features in modality m, as in typical MKL. We restrict the model space by using the same kernel function (e.g., an RBF kernel) for each modality. The modality kernels are combined into a single kernel, $k_{\boldsymbol{\eta}}$, in a convex combination parameterized by the kernel weighting vector, $\boldsymbol{\eta}$. Let $\mathbf{x}_i^{(m)}$ be the ith feature vector that contains only the features belonging to modality m, and M be the total number of modalities. Then $k_{\boldsymbol{\eta}}$ is defined as follows:

$$k_{\boldsymbol{\eta}}(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\eta}) = \sum_{m=1}^{M} \boldsymbol{\eta}_m k_m(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)})$$

such that $\boldsymbol{\eta}_m > 0, m = 1, \ldots, M$ and $\sum_{m=1}^{M} \boldsymbol{\eta}_m = 1$. Thus the LSSVM-based MKL model can be learned using the same optimization as the LSSVM with the additional constraint of the convex combination of kernel weights $\boldsymbol{\eta}$.

When multiple tasks are learned at the same time in MTMKL, each task t has its own vector of kernel weights, $\boldsymbol{\eta}^{(t)}$, which are regularized globally by a function which penalizes divergence from the weights of the other tasks. This allows information about the kernel weights to be shared between tasks so that each task benefits from the data of other tasks. In particular, if the model is highly regularized, then the kernel weight on the mth modality (i.e., $\boldsymbol{\eta}_m^{(t)}$) will be very similar across all tasks t. As such, each task will treat the modalities as having similar importance. Note that even though the kernel weights might be highly regularized, the task-specific models can still learn a diverse set of decision boundaries within the same kernel space.

The optimal $\boldsymbol{\eta}^{(t)}$ for all tasks $t = 1, \ldots, T$ can be learned by solving a min-max optimization similar to the LSSVM-based MKL model, but with the addition of the regularization function, $\Omega(\{\boldsymbol{\eta}^{(t)}\}_{t=1}^{T})$. A weight ν placed on the regularization function $\Omega(\cdot)$ controls the importance of the divergence. When $\nu = 0$ the tasks are treated independently, and as ν increases, the task weights are increasingly restricted to be similar.

For simplicity of notation we denote the objective function for a single task's LSSVM-based MKL model as follows:

$$J^{(t)}\left(\alpha^{(t)}, \boldsymbol{\eta}^{(t)}\right) = -\frac{1}{2}\sum_{i=1}^{N^{(t)}}\sum_{j=1}^{N^{(t)}} \alpha_i^{(t)}\alpha_j^{(t)} y_i y_j k_{\boldsymbol{\eta}}^{(t)}(x_i, x_j) - \frac{1}{2C}\sum_{i=1}^{N^{(t)}} \alpha_i^2 + \sum_{i=1}^{N^{(t)}} \alpha_i$$

where the superscript (t) denotes the parameters or functions specific to task t.

Thus, all of the parameters of the LSSVM-based MTMKL model can be learned by solving the following min-max optimization problem:

$$\begin{aligned}
&\underset{\{\boldsymbol{\eta}^{(t)}\}_{t=1}^{T}}{\text{minimize}}\ \underset{\{\boldsymbol{\alpha}^{(t)}\}_{t=1}^{T}}{\text{maximize}} && \nu\Omega\left(\{\boldsymbol{\eta}^{(t)}\}_{t=1}^{T}\right) + \sum_{t=1}^{T} J^{(t)}\left(\alpha^{(t)}, \boldsymbol{\eta}^{(t)}\right) \\
&\text{subject to} && \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& && \sum_{m=1}^{M} \boldsymbol{\eta}_m^{(t)} = 1, t = 1, \ldots, T \\
& && \boldsymbol{\eta}_m^{(t)} \geq 0, \forall m, \forall t
\end{aligned}$$

The iterative gradient descent method proposed by Kandemir et al. (2014) is used to train the model given an initial set of model parameters. The method alternatively (1) solves a LSSVM for each task given

$\boldsymbol{\eta}^{(t)}$ and (2) updates $\boldsymbol{\eta}$ in the direction of negative gradient of the joint objective function (see Algorithm 4).

Let the joint objective function be $O_{\boldsymbol{\eta}}$. We write the gradient as follows:

$$\frac{\partial O_{\boldsymbol{\eta}}}{\partial \boldsymbol{\eta}_m^{(t)}} = \nu \frac{\partial}{\partial \boldsymbol{\eta}_m^{(t)}} \Omega\left(\{\boldsymbol{\eta}^{(t)}\}_{t=1}^{T}\right) - \frac{1}{2} \sum_{i=1}^{N^{(t)}} \sum_{j=1}^{N^{(t)}} \boldsymbol{\alpha}_i^{(t)} \boldsymbol{\alpha}_j^{(t)} y_i^{(t)} y_j^{(t)} k_m(\boldsymbol{x}_i^{(m)}, \boldsymbol{x}_j^{(m)})$$

Algorithm 4: MTMKL Algorithm

```
Initialize $\boldsymbol{\eta}^{(t)}$ as $(1/T, ..., 1/T)$, $\forall t$ **while** *not converged* **do**

    **end**
    Solve each LSSVM-based MKL model using $\boldsymbol{\eta}^{(t)}$, $\forall t$
Update $\boldsymbol{\eta}^{(t)}$ in the direction of $-\partial O_{\boldsymbol{\eta}} / \partial \boldsymbol{\eta}^{(t)}$, $\forall t$
```

Following Kandemir et al. (2014), we use two different regularization functions. The first, $\Omega_1(\cdot)$, penalizes the negative total correlation, as measured by the dot product between the two kernel weight vectors $< \boldsymbol{\eta}^{(t_1)}, \boldsymbol{\eta}^{(t_2)} >$:

$$\Omega_1(\{\boldsymbol{\eta}^{(t)}\}_{t=1}^{T}) = -\sum_{t_1=1}^{T} \sum_{t_2=1}^{T} < \boldsymbol{\eta}^{(t_1)}, \boldsymbol{\eta}^{(t_2)} >$$

The second regularization function, $\Omega_2(\cdot)$, penalizes the distance of kernel weights in Euclidean space:

$$\Omega_2(\{\boldsymbol{\eta}^{(t)}\}_{t=1}^{T}) = \sum_{t_1=1}^{T} \sum_{t_2=1}^{T} ||\boldsymbol{\eta}^{(t_1)} - \boldsymbol{\eta}^{(t_2)}||_2$$

8.5.3 Hierarchical Bayesian Logistic Regression (HBLR)

The methods we have presented so far rely on clustering participants *a priori* based on their personality and demographics, in order to build a robust model that can generalize to new people. However, it would be preferable if we could train a model to automatically cluster participants, not based on characteristics we *assume* to be related to mood prediction, but instead directly using the unique relationship each person has between their physiology, behavior, the weather, and their resulting mood. As mentioned previously, individuals may be affected very differently by the same stimuli; e.g., one person may become more calm when the weather is rainy, while another may become annoyed. The ability to group individuals based on these differing reactions could thus be highly valuable.

Therefore, we now consider a non-parametric hierarchical Bayesian model which can implicitly learn to cluster participants that are most similar in terms of their relationship between the input features and their resulting mood. Further, the model learns a soft clustering, so that a participant does not need to be assigned to a discrete, categorical cluster, but rather can belong to many clusters in varying degrees.

In hierarchical Bayesian MTL approaches, the model for each task draws its parameters from a common prior distribution. As the model is trained, the common prior is updated, allowing information to be shared across tasks. The model we adopt, which was originally proposed by Xue et. al. (Xue et al., 2007), draws logistic regression (LR) weights for each task from a shared Dirichlet Process (DP) prior; we call this model Hierarchical Bayesian Logistic Regression (HBLR).

In contrast with our prior approaches (MTL-NN and MTMKL), the HBLR model allows us to directly define each task as predicting the wellbeing of a single person, since the model is able to implicitly learn its own clustering over people. While the implicit clustering provides valuable insights into groups of people that have a different relationship between their physiology, behavior, and wellbeing, it also means that HBLR cannot make predictions about a new person's mood without first receiving at least one labeled training data point from that person. Still, HBLR can quickly be adapted to make predictions about a new person (Xue et al., 2007), and the predictions will improve with more data.

The implicit clustering mechanism is accomplished through the choice of the Dirichlet Process prior. The DP prior induces a partitioning of the LR weights into K clusters, such that similar tasks will end up sharing the same weights. Specifically, for each task t, the model parameters $w^{(t)}$ are drawn from a common prior G which is sampled from a DP:

$$\begin{aligned} w^{(t)}|G &\sim G, & \alpha &\sim Ga(\tau_1, \tau_2) \\ G &\sim DP(\alpha, G_0), & G_0 &\sim N_d(\mu, \Sigma) \end{aligned}$$

where Ga is a Gamma distribution and N_d is a $d-$dimensional multivariate normal distribution. The distribution G_0 is the base distribution and represents our prior belief about the distribution from which the weights are drawn. Following (Xue et al., 2007), we set $\mu = \mathbf{0}$ and $\Sigma = \sigma \mathbf{I}$, which reflects the prior belief that the weights should be uncorrelated and centered around zero (equally likely to be positive or negative). Here, σ is a hyperparameter. The scaling or innovation parameter of the DP $\alpha > 0$ affects the likelihood that a new cluster will be generated; as α decreases the weights generated by the DP will become more concentrated around only a few distinct clusters. In this case, α is distributed according to a diffuse prior represented by a Gamma distribution with hyperparameters τ_1 and τ_2.

The goal of the HBLR model is to learn a posterior distribution over the variables defined above given the observed data. When each task is defined as learning the decision boundary for a single person, learning the posterior allows the model to:

(a) learn a non-parametric clustering of similar people

(b) perform MTL by jointly learning logistic regression classifiers for each cluster.

Here, we define people as similar when the classification boundaries of their wellbeing prediction tasks are close; that is, when their respective weight vectors are similar. This implies that similar people have a similar relationship between their input features and their resulting wellbeing.

Learning the complete posterior distribution is intractable, so mean-field variational Bayesian inference (VI) is used to approximate the true posterior; the VI equations are derived by Xue et al. (2007). The variational approximation of the posterior contains three sets of parameters that the model must learn. The first is a matrix $\Phi \in \mathbb{R}^{T \times K}$, where T is the number of tasks (or participants), and K is the number of clusters. The Φ is essentially the learned soft clustering of users (see (a) above); each row $\phi^{(t)} \in R^K$ represents the degree to which person t belongs to each of the K clusters. Although the non-parametric nature of the model could theoretically allow for an infinite number of clusters, there is no loss in generality if K is limited to the number of tasks in practice. We make an additional computational enhancement to the algorithm by removing clusters for which all entries of ϕ_k are less than machine epsilon, which allows for faster convergence.

The second set of parameters are (θ_k, Γ_k) for $k = 1, \ldots, K$, which parameterize a unique distribution over the LR weights for each of the K clusters (see (b) above). That is, each cluster k draws its weights from a multivariate normal distribution as follows:

$$w_k \sim N_d(\theta_k, \Gamma_k), k = 1, \ldots, K$$

Note that in expectation (θ_k, Γ_k) center around the μ and Σ parameters of the base distribution.

To learn all the parameters, we use a coordinate ascent algorithm developed by Xue et. al. (Xue et al., 2007). The parameters $(\Phi, \{\theta_k\}_{k=1}^K, \{\Gamma_k\}_{k=1}^K)$ are initialized to their respective uniform priors; that is, each task having equal contribution to each cluster to initialize Φ and setting θ_k and Γ_k to μ and Σ for each k. Each parameter is then iteratively re-estimated until convergence.

To predict a new test sample $x_*^{(t)}$, we would ideally like to use the following equation, where we integrate over the learned distribution on the classifier's weights:

$$\begin{aligned} p(y_*^{(t)} = 1 | x_*^{(t)}, &\Phi, \{\theta_k\}_{k=1}^K, \{\Gamma_k\}_{k=1}^K) \\ &= \sum_{k=1}^{K} \phi_k^{(t)} \int \sigma(w_k^{*T} x_*^{(t)}) N_d(\theta_k, \Gamma_k) dw_k^* \end{aligned}$$

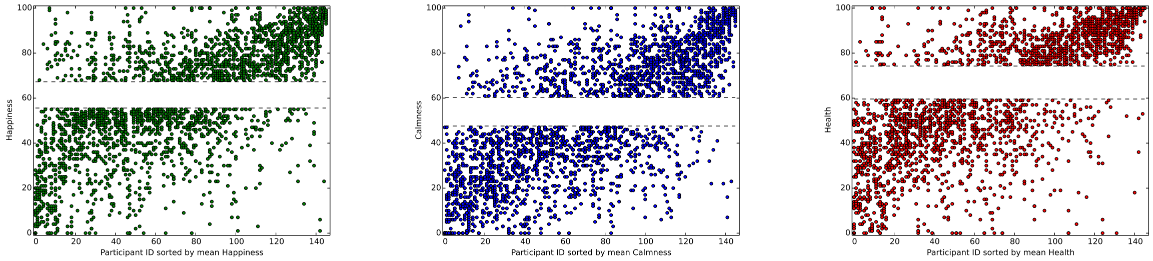
where σ is the sigmoid function of a typical LR classifier.

However, computing this integral is intractable. Therefore, the prediction function uses an approximate form of the integral derived in (MacKay, 1992):

$$p(y_*^{(t)} = 1 | x_*^{(t)}, \Phi, \{\theta_k\}_{k=1}^K, \{\Gamma_k\}_{k=1}^K) \approx \sum_{k=1}^{K} \phi_k^{(t)} \sigma \left(\frac{\theta_k^T x_*^{(t)}}{\sqrt{1 + \frac{\pi}{8} x_*^{(t)T} \Gamma_k x_*^{(t)}}} \right)$$

8.5.4 *Multi-task learning experiments*

In this second set of experiments, we attempt to predict students' next-day wellbeing in terms of their happiness, stress, and health. Each morning and evening, participants self-reported their mood (sad/happy), stress (stressed out/calm), and health (sick/healthy) on a visual analog scale from 0-100. Table 8.1 shows the correlation between these metrics. To create binary classfification labels, these scores are split based on the median value. In the previous section



(a) Mood

(b) Stress

(c) Health

Figure 8.6: Distribution of self-report labels after discarding the middle 20%. Participants are listed on the x-axis, in order of their average self-report value for that label (each participant is one column). Almost all participants have data from both label classes.

and in other prior work, we relied on discarding the most neutral scores before creating binary labels in order to disambiguate the classification problem; i.e., the middle 40% of scores were discarded due to their questionable nature as either a 'happy' or 'sad' state (Jaques et al., 2015a,b). We instead make the problem decidedly harder by discarding only the middle 20% of scores. We also discard participants with less than 10 days worth of data, since they could not provide enough data to train viable models. The resulting dataset comprises 104 users and 1842 days.

Figure 8.6 shows the raw values reported for mood, stress, and health for each participant after the middle 20% of scores have been removed. Points appearing above the removed values are assigned a positive classification label, while points below are assigned a negative label. As is apparent from the figures, although some participants predominantly report one label class almost all participants' reports span the two classes. This implies that the need for personalization is not simply due to the fact that some participants are consistently sad while some are consistently happy, for example. Personalization is required because people react differently to very similar stimuli, and a single, impersonal classifier cannot capture these differences.

To assess whether personalization via MTL provides significant performance benefits, we compare it to two other approaches. First, we compare each algorithm to its single task learning (STL) equivalent. HBLR is compared to conventional LR, MTMKL to LSSVM, and MTL-NN to a generic NN[1]. Second, to determine whether personalization via MTL has a performance advantage over simply using MTL itself, we also explore multitasking over the related wellbeing measures; in other words, in this condition we treat predicting mood, stress, and health as related tasks. Note that this moods-as-tasks approach to MTL is similar to that taken in prior work (e.g. Kandemir et al. (2014); Jaques et al. (2015b); Xia and Liu (2015); Zhang et al.

[1] Note that the dataset and classification labels used for these experiments is different than that of the previous section, so we cannot directly compare the STL results.

(2017)).

To create the datasets used for training the models and avoid testing data contamination, a random 80/20% split was used to partition the SNAPSHOT data into a train and test set. We then apply 5-fold cross validation to the training set to test a number of different parameter settings for each of the algorithms described in Section 8.5. Finally, each model is re-trained using the optimal parameter settings, and tested once on the held-out testing data; the test performance is reported in the following section.

To tune the neural networks, we consistently used learning rate decay and the Adam optimizer (Kingma and Ba, 2014), and tuned the following settings: the number and size of hidden layers, batch size, learning rate, whether or not to apply dropout, and the L2 β weight. Based on previous work that has successfully trained MTL NNs with few samples (Jin and Sun, 2008), we choose a simple, fully-connected design with 2-4 hidden layers. For HBLR, we tuned the τ_1, τ_2, and σ parameters, while for MTMKL we tuned C, β, the type of kernel (linear vs. radial basis function (RBF)), the type of regularizer function ($\Omega_1(\cdot)$ vs $\Omega_2(\cdot)$), and ν. For MTMKL we also define the following modalities: classifier, location, survey interaction, survey activities, survey sleep, weather, call, physiology from 3am to 10am, screen, and SMS. More detail on these modalities is provided in Table 8.5.

All of the code for the project, which is written in Python and TensorFlow (Abadi et al., 2015), has been released open-source at: `https://github.com/mitmedialab/personalizedmultitasklearning`

8.5.4.1 Feature selection

Since the dataset is small, we apply feature selection to reduce the chance of overfitting. While there are many ways to do this, in this work features are selected based on assessing ANOVA F-scores between each feature and the classification label using the training data and removing highly correlated features, with the constraint that at least one feature from each of the above data sources is retained. This process gave rise to a total of 21 features, which are listed in Table 8.5.

Modality	Features
Classifier	Day of the week
Physiology 3am-10am	% mins with >= 5 SCRs (w/o artifacts)
	Temperature weighted SCR AUC
Location	Time on campus
	Log likelihood of day given previous days
Call	Total missed calls
SMS	Total incoming (midnight-3am)
	Number of unique contacts outgoing
	Number unique incoming (5-11:59pm)
	Number unique outgoing (5-11:59pm)
Screen	Total duration (Midnight-3am)
	Total number on/off events (5-11:59pm)
Survey Activities	Exercise duration
	Study duration
Survey Interaction	Positive social interaction
	Presleep in-person interaction (T/F)
Survey Sleep	Number of naps
	All-nighter (T/F)
Weather	Cloud cover rolling std. dev.
	Max precipitation intensity
	Pressure rolling std. dev.

Table 8.5: Features selected from each modality for the second set of experiments. There are a total of 21.

8.5.4.2 Method for analyzing HBLR clusters

Because the clusters learned by the HBLR model may be fundamentally different than those that can be obtained using other methods, we are interested in defining a way to analyze which type of participants are represented within each cluster. For example, does a certain cluster tend to contain participants that have a significantly higher trait anxiety score (as measured by the pre-study survey)?

The analysis is complicated by the fact there is no discrete assignment of participants to clusters; rather, a participant may have some degree of membership in many or all of the clusters, as defined by $\phi^{(t)}$. To solve this issue, we first define a matrix $\mathbf{P} \in \mathbb{R}^{T \times M}$, where T is the number of participants and M is the number of pre-study measures (such as Big Five personality, PSS, etc.). Thus, $P_{t,m}$ represents person t's score on measure m. Using $\mathbf{P}$, we can then compute a score representing the average value of each pre-study measure for each cluster, as follows:

$$\boldsymbol{Q}_{k,m} = \frac{\sum_t \boldsymbol{P}_{t,m} \phi_k^{(t)}}{\sum_t \phi_k^{(t)}}$$

where $\boldsymbol{Q} \in \mathbb{R}^{K \times M}$ and K is the number of clusters learned by the HBLR model. $\boldsymbol{Q}_{k,m}$ can be considered a weighted average of a cluster's pre-study trait, where the weights are the degree of membership of each participant in that cluster.

To test whether a cluster's $\boldsymbol{Q}_{k,m}$ value is significantly different than the group average, we use a one-samples t-test to compare $\boldsymbol{Q}_{k,m}$ to the values for measure m reported by participants on the pre-study survey. We apply a Bonferroni correction based on the number of comparisons made across the different clusters within each outcome label (i.e. mood, stress, health).

8.5.5 Multi-task learning results and discussion

	Classifier	**Mood**	**Stress**	**Health**
Baseline	Majority class	50.4%, .500	50.7%, .500	54.4%, .500
STL	LSSVM	60.2%, .603	58.1%, .581	62.3%, .614
	LR	56.9%, .569	59.4%, .594	55.4%, .544
	NN	60.5%, .606	60.1%, .600	65.9%, .648
	NN (all feats)	65.8%, .658	67.9%, .678	59.0%, .591
MTL - moods	MTMKL	59.4%, .594	58.8%, .587	62.0%, .610
	HBLR	58.3%, .583	57.8%, .578	55.1%, .551
	MTL-NN	60.2%, .602	60.1%, .600	65.3%, .643
	MTL-NN (all feats)	67.0%, .670	68.2%, .682	63.0%, .623
MTL - people	MTMKL	**78.7%, .787**	**77.6%, .776**	**78.7%, .786**
	HBLR	**72.0%, .720**	**73.4%, .734**	**76.1%, .760**
	MTL-NN	**77.6%, .776**	**78.6%, .785**	**79.7%, .792**
	MTL-NN (all feats)	**78.4%, .784**	**81.5%, .815**	**82.2%, .818**

Table 8.6: Prediction performance (Accuracy and AUC) of the STL, MTL-moods, and MTL-user methods. Bolded entries represent significant improvements over the STL model, indicating that multitasking for personalization is by far the most effective approach.

The accuracy and Area Under the ROC Curve (AUC) of each of the wellbeing prediction models is shown in Figure 8.7 and Table 8.6, along with the majority class baseline (the peformance that can be expected from simply predicting the most frequent label in the training data). For most models, we found that using feature selection improved performance. Since NNs often benefit from large input vectors, we tested the performance of the MTL-NN on the full set of 343 features as well, and include these results in Table 8.6.

As is evident from Table 8.6 and Figure 8.7, the accuracy obtained using traditional STL ML classifiers is poor, reaching a maximum of only 60-66%; this is similar to prior work that trained STL classifiers to detect mood on a simplified version of this dataset (Jaques et al., 2015a). The performance obtained with the three MTL models when multitasking over the related outcome labels, i.e. mood, stress, and

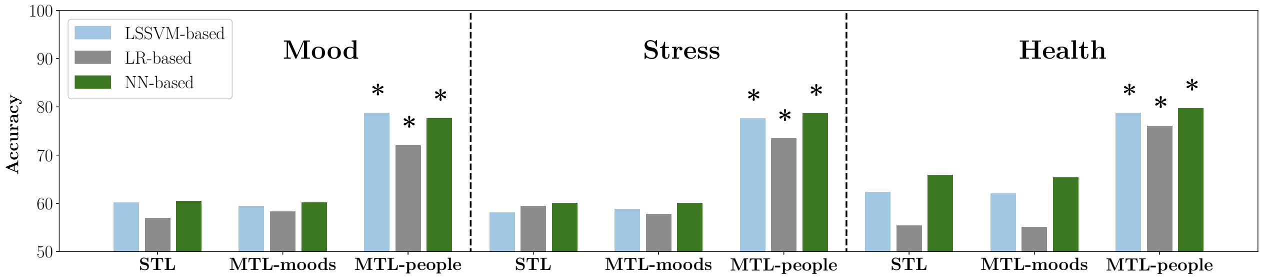
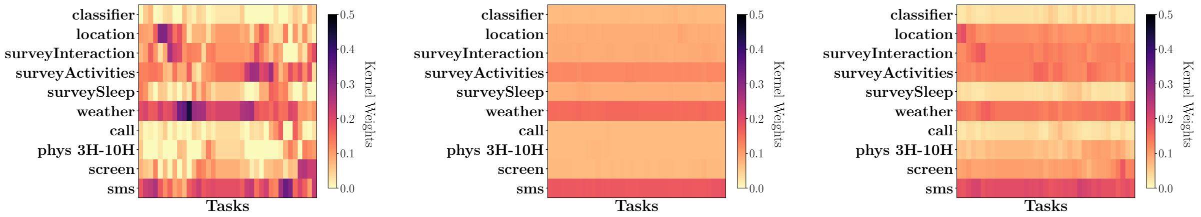


Figure 8.7: Accuracy for each type of model in the STL, MTL-moods, and MTL-people approaches. Note that the accuracy significantly ($* = p < 0.05$) improves when using multitasking over people for each label and for each machine learning method tested.

health is shown as *MTL - moods*. Evidently, multitasking in this way does not significantly enhance performance. This could be because the outcome labels are not sufficiently related to each other to benefit from sharing parameters, or that the regularization imposed by MTL limits the models' capacity more than it benefits the generalization performance. Therefore, it is clear that at least for this data, MTL alone is not sufficient to improve mood prediction classifiers.

Rather, it is using MTL to account for individual differences that is important. As is clear from both Table 8.6 and Figure 8.7, using MTL to personalize ML models by multitasking over clusters of similar people provides dramatic improvements to mood prediction performance. The improvement in accuracy over the non-personalized models ranges from 11-21%. McNemar tests of the predictions with a Bonferroni correction applied within each label type revealed that the personalized models significantly outperformed ($p < .05$) both the STL and *MTL - moods* approaches, over all model and label types. These scores represent state-of-the-art performance on this dataset, surpassing prior published work by 5-13% prediction accuracy (Jaques et al., 2015a,b).

Given the impressive performance of the personalized MTL models, in the following sections we focus on analyzing the weights and clusters learned by the personalized MTMKL and HBLR models, both of which can help provide important insights into how the wellbeing of different groups of people is affected by their physiology, their behavior, their use of technology, and the weather.



(a) Mood ($\nu = 0.1$)

(b) Stress ($\nu = 100$)

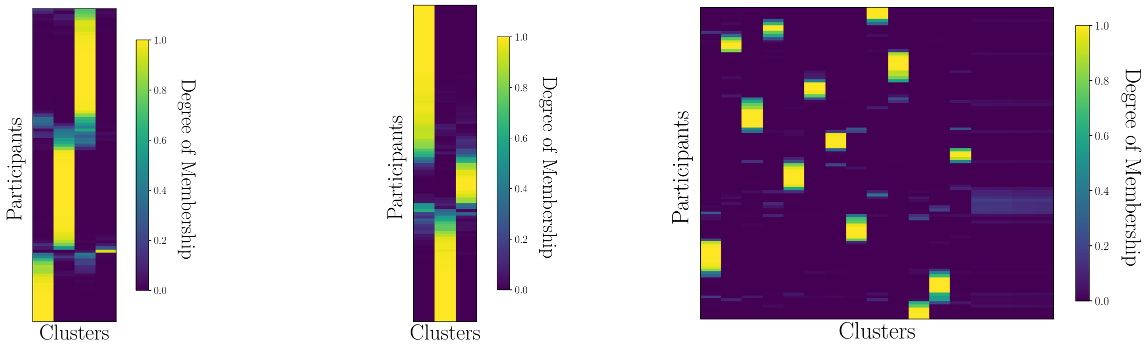
(c) Health ($\nu = 1$)

Figure 8.8: MTMKL kernel modality weights, reflecting which feature type is most important to the classifier for each task. The ν parameter controls how heavily the task weights are regularized to be similar, and was set by the hyperparameter search. People are highly individualistic in how strongly their mood is affected by different modalities like weather.

8.5.5.1 MTMKL

The MTMKL model learns a weighting over the 10 modalities for each task. As described in Section 8.5.2, the ν parameter controls how strongly the tasks' weights are constrained to be similar. Figure 8.8 shows the weights learned by the personalized MTMKL model for each outcome label and each cluster of people. The figure demonstrates that for predicting stress, the hyperparameter search selected a large value of ν, which constrained the kernel weights to be highly similar across all clusters. However, for mood the value of ν was much smaller, resulting in more diverse kernel weights for the different tasks. This could suggest that there is more individual variability in how well these features can predict mood, while the relationship to stress is more consistent. It appears that for mood prediction, differences in features like weather, location, and screen are important for some types of people but not others.

The modality weights learned by MTMKL can potentially provide interesting insights for designing a mood prediction system. For example, we see that for this set of features, overall the differences in weather, SMS, and survey (representing exercise and studying duration) tend to be more informative. This may suggest that not only is it important to include weather in a mood prediction system, but that developing ways to automatically and unobtrusively detect when a participant is exercising or studying could be a valuable time investment. Further, we see that the call features tend to be less informative compared to the other features, so perhaps it is not necessary to monitor participants' call patterns to predict wellbeing. Removing this form of data collection could potentially enhance privacy for participants in the SNAPSHOT study.



(a) Mood

(b) Stress

(c) Health

Figure 8.9: Resulting soft clustering (Φ) when predicting the different labels (mood, stress, and health). Each row shows one of the 104 participant's degree of membership in each cluster. We note that there were 4,3, and 17 clusters needed in predicting happiness, stress, and health, respectively.

8.5.5.2 HBLR

The HBLR model learns a non-parametric soft clustering of participants based on the relationship between their input features and resulting mood. Figure 8.9 shows the clustering learned for predicting each of the three outcome labels, where the intensity of the color represents the degree to which each participant belongs to each cluster. The number of clusters which had at least one participant with a degree of membership exceeding machine epsilon were 4, 3, and 17 for the mood, stress, and health prediction models, respectively. However, this does not imply that there are only three types of people which have a different relationship between the features and stress. Because of the soft clustering, a given person can belong to many clusters and thus combine the decision boundaries learned for each, as explained below.

As discussed previously, each cluster in the HBLR model learns a multivariate normal distribution over the weight vector w_k^*. In Figure 8.10 we show examples of the different marginal distributions learned over a single feature (total number of screen on events (5pm-midnight)) for the four mood clusters. We note that for these two features, cluster 0 and cluster 1 have very different distributions on the LR weights. For example, in Figure 8.10 we see that cluster 0 places a negative weight on the feature whereas cluster 1 places a positive weight on the same feature. Thus, when participants who belong almost exclusively to cluster 0 use their phone excessively in the evening, the model will be more likely to predict a sad day tomorrow. In contrast, the model is more likely to predict a happy day tomorrow for participants belonging almost exclusively to cluster 1 based on the same behavior.

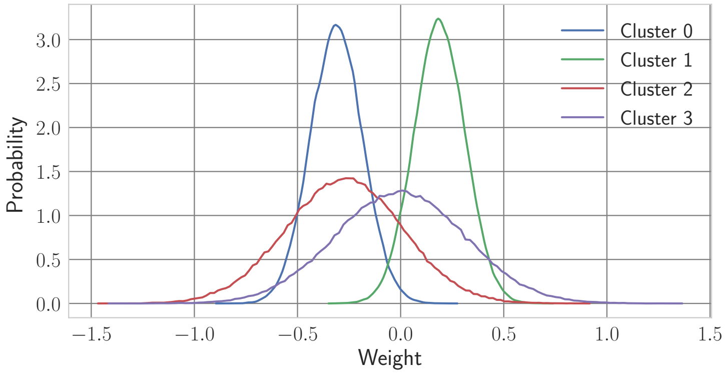


Figure 8.10: Distribution of HBLR weights on the *total number of screen on events (5pm-midnight)* feature for each cluster when predicting tomorrow's mood

However, because participants do not belong exclusively to one cluster or another, the marginal distribution over a weight parameter for a given participant can be more complex than a multivariate normal. For example, Figure 8.11 shows an example of the weight distributions for 3 different participants. For Participant 5, the model has constructed a bimodal distribution over the weight by combining the distributions of multiple clusters. Thus, the model is able to customize the decision boundary for each person while still clustering the participants into similar archetypes.

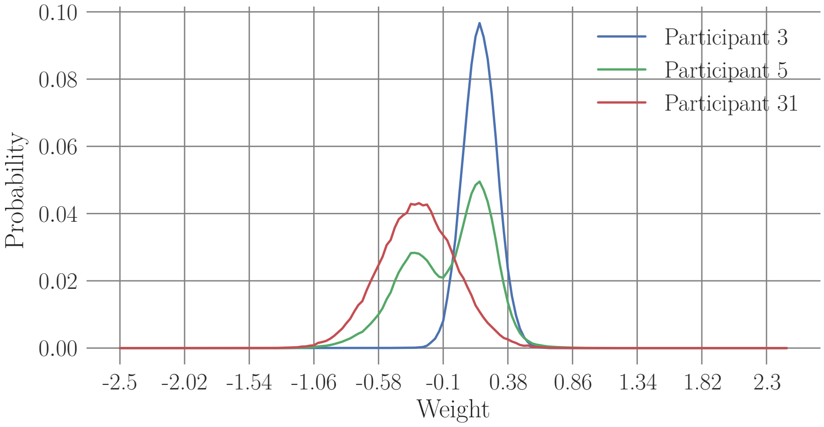


Figure 8.11: Example of different weight distributions induced by the soft clustering for 3 different participants in the mood prediction. Participant 3 is almost exclusively in cluster 1, participant 5 is has membership in clusters 0, 1, and 2, and participant 31 is almost exclusively in cluster 2.

As described in Section 8.5.4.2, we would like to determine if the clusters learned by the HBLR model differ significantly in terms of the typical personality or mental health scores of the participants. Following the procedure outlined in that section, we computed the average scores for each cluster on each of the pre-study trait measures (i.e. the matrix **Q**), then conducted a limited number of significance tests with a Bonferroni corrcection to determine if there were significant differences among the clusters for some of the traits. Since the HBLR clustering is based on latent factors underlying the data that are unknown before training, it is not possible to determine *a prior* what traits may be particularly relevant to a given cluster. Below, we discuss the results of these computations for some notable traits of the mood and stress clusters. We do not show the same analysis for health, since the 17 different clusters in the health model render it impractical to present the results.

Table 8.7 shows the relevant trait values for the mood clusters, including the average value for those traits computed over all participants in the study. According to these findings, the clusters learned by the HBLR model in predicting mood can be characterized as a) Judging and Sensing personality types; b) people with better than average sleep quality (PSQI); c) Agreeable people, and d) happy Extraverts with low state and trait anxiety. This could suggest that these traits are highly relevant for predicting how a person's mood will change given the input features. For example, since poor sleep quality has been shown to have a negative effect on mood (Bower et al., 2010), perhaps the normally high sleep quality of participants in cluster 1 makes their mood more sensitive to sleep disturbances.

Cluster	**Pre-study measure**	**All participants**	**Cluster $Q_{k,m}$**	t	p
0	Percent happy days	$M = 49, SD = 37$	56	-1.86	$> .10$
0	**Judging**	$M = 61, SD = 21$	73	-7.69	$< .001$
0	**Sensing**	$M = 47, SD = 20$	57	-7.22	$< .001$
1	Percent happy days	$M = 49, SD = 37$	55	-1.81	$> .10$
1	**PSQI**	$M = 4.7, SD = 2.3$	4.1	3.48	$< .01$
2	Percent happy days	$M = 49, SD = 37$	41	2.29	$> .10$
2	**Agreeableness**	$M = 50, SD = 28$	43	3.63	$< .01$
3	**Percent happy days**	$M = 49, SD = 37$	78	-8.00	$< .001$
3	**Extraversion**	$M = 49, SD = 30$	76	-13.1	$< .001$
3	**State anxiety**	$M = 38, SD = 10$	30	10.9	$< .001$
3	**Trait anxiety**	$M = 43, SD = 10$	36	9.85	$< .001$

Table 8.7: Post-hoc analysis of the clusters learned for the HBLR mood model, using pre-study personality and trait measures. Bolded entries represent significant differences from the sample average.

It is particularly interesting to relate these results to the average value for the weights learned for these clusters, as shown in Figure 8.12. For example, it appears that the "Agreeable" cluster (cluster 2) places high weight on four social interaction features; this is consistent with research indicating that those with an Agreeable personality type value getting along with others (John and Srivastava, 1999). In contrast with this cluster, the "High sleep quality" cluster (cluster 1) places negative weight on features related to SMS use in the evening. Finally, we observe that the "Judging and Sensing" cluster (cluster 0) has a positive association with exercise, but a negative association with time spent on campus.

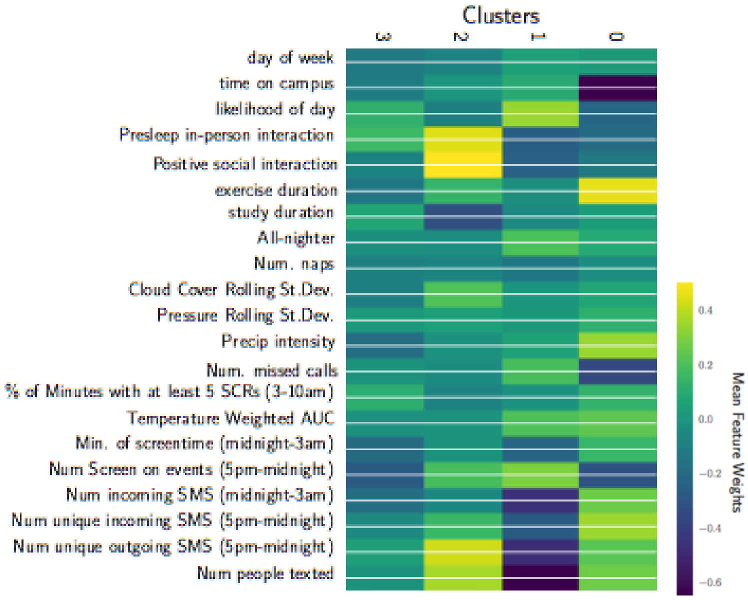


Figure 8.12: Mean feature weights for mood clusters in HBLR model. The positive label is "Happy" so features with positive (negative) mean weights contribute to being more happy (sad) tomorrow.

Note that we also examined whether the HBLR model would simply cluster participants with a tendency to be particularly happy or particularly sad together, in order to more easily make accurate predictions. As shown in Table 8.7, three of the clusters do not differ significantly from the group average in terms of average percent of happy days, although cluster 3 (Extroverts with low state and trait anxiety) does correspond to particularly happy participants.

The results of the same analysis of HBLR cluster pre-study measures for the stress model are shown in Table 8.8. In this case, none of the clusters differed significantly from the group average in terms of the percentage of calm days. While we did not detect any significant differences from the group average for cluster 0, cluster 1 represents an intuitively salient group: conscientious people with a high GPA. It makes sense that this clustering would be relevant to predicting stress, since conscientious students who are concerned about their grades are likely to have strong stress reactions in an academic environment. As shown in Figure 8.13 this cluster places a positive weight on the "likelihood of day" feature, which is a measure of how routine the participants location patterns were that day, and will be higher if the participant travels mainly to typical work and home locations. Stress cluster 2 represents students who are extraverted, with slightly increased BMI and lowered physical health. In examining Figure 8.13, we can see that cluster 2 has highly positive mean feature weights on the SMS features, which is consistent with the trait of Extraversion. On the contrary, cluster 1 has highly negative weights on the social SMS features, meaning more SMS use for these participants would increase the likelihood of predicting a stressful day tomorrow. One of several possible explanations is that perhaps these conscientious, high GPA students become stressed by having to balance their academic goals and social life.

Cluster	Pre-study measure	All participants	Cluster $Q_{k,m}$	t	p
0	Percent calm days	$M = 48, SD = 38$	46	.492	$> .60$
1	Percent calm days	$M = 48, SD = 38$	55	-1.88	$> .10$
1	**GPA**	$M = 4.4, SD = .61$	4.6	-3.95	$< .001$
1	**Conscientiousness**	$M = 51, SD = 28$	58	-3.43	$< .01$
2	Percent calm days	$M = 48, SD = 38$	39	2.32	$> .10$
2	**Extraversion**	$M = 49, SD = 30$	58	-4.50	$< .001$
2	**BMI**	$M = 24, SD = 4.4$	25	-4.09	$< .001$
2	**PCS**	$M = 58, SD = 4.2$	57	3.77	$< .01$

Table 8.8: Computed pre-study measures for the HBLR stress prediction clusters. Bolded entries represent significant differences from the sample average.

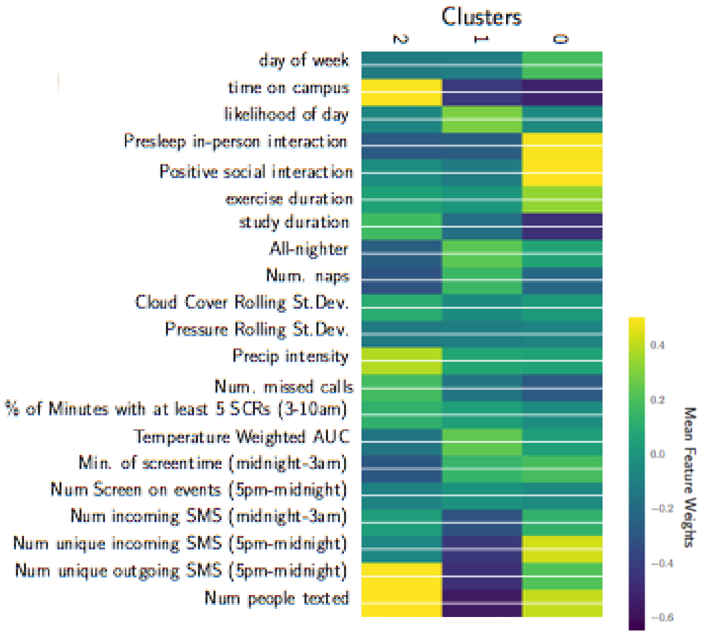


Figure 8.13: Mean feature weights for clusters in HBLR model. For stress the positive label is “Calm” so features with positive (negative) mean weights contribute to being more calm (stressed) tomorrow.

8.6 *Personalized regression*

Treating wellbeing as a binary state (e.g., happy vs. unhappy) is a limitation of the previous results. Such a coarse approach could miss important distinctions relevant for clinical applications. For example, the lowest possible mood score is treated as equivalent to a slightly lower than average score, while it is possible that only the former is a clinically significant sign of depression. Being able to directly predict a fine-grained estimate of mood, rather than a binary category, could be extremely valuable. Further, since the system relies on continuous collection of data from many sources in real-world, daily life settings, the data will inevitably contain noise. A robust ML system that can provide an estimate of the degree of uncertainty for a given mood prediction is therefore highly desirable.

To resolve these issues, in this section we train DNNs and Gaussian Processes (GPs) to simultaneously predict tomorrow's mood, health and stress intensity from the same data. Multi-task learning (MTL) is then in the same manner to train a personalized DNN. We use a Domain Adaptation (DA) approach to customize GPs to the individual, adjusting the models to each person by updating the posterior distribution of the GP – a method described in detail in the next section. Empirical results demonstrate that the proposed personalization results in considerable performance boost. To the best of our knowledge, this is the first personalized approach for automatic prediction of fine-grained self-reported mood and wellbeing levels. These personalized predictions of future mood and wellbeing have potential to meaningfully improve real-world monitoring and intervention applications.

8.6.1 Gaussian Processes for personalized domain adaptation

We consider a supervised setting for domain adaptation, where we are given a relatively large amount of labeled training data (*source* domain), and a considerably smaller set of labeled data in the *target* domain. Furthermore, we assume a person-dependent setting, i.e., the (non-overlapping) data of target persons are available in the source and target domain. Thus, our goal is to learn a general prediction model from data of all persons, and then leverage the limited data of a target person to perform the model adaptation to that specific person. Formally, let $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ be the input (features) and output (labels) spaces, respectively. We assume that the input space is composed of the source and target domains, $\mathcal{S}$ and $\mathcal{T}$, respectively, that may differ in feature distribution. Hence, $\boldsymbol{X}^{(s)} = \{\boldsymbol{x}^{(s)}_{n_s}\}^{N_s}_{n_s=1}$ and $\boldsymbol{X}^{(t)} = \{\boldsymbol{x}^{(t)}_{n_t}\}^{N_t}_{n_t=1}$, with $\boldsymbol{x}^{(s)}_{n_s}, \boldsymbol{x}^{(t)}_{n_t} \in \mathbb{R}^D$, and $N_t \ll N_s$. Similarly, $\boldsymbol{Y}^{(s)} = \{\boldsymbol{y}^{(s)}_{n_s}\}^{N_s}_{n_s=1}$ and $\boldsymbol{Y}^{(t)} = \{\boldsymbol{y}^{(t)}_{n_t}\}^{N_t}_{n_t=1}$ are the output labels for the source and target domains, respectively, where $\boldsymbol{y}^{\{s,t\}}_n$ represents the intensity level of the wellbeing dimension that we wish to estimate (i.e., mood, health, or stress).

8.6.1.1 Gaussian Processes (GPs)

Here, we introduce briefly the modeling framework of GP regression, that we use as the base model in our personalization approach via domain adaptation. We employ GPs for two reasons: (i) it is a non-parametric model, which allows us to efficiently capture non-linear relationships between input features and output labels using kernel functions; and (ii) due to its probabilistic nature, the model adaptation can be performed in a principled manner by deriving a posterior distribution conditioned on the adaptation (target) data. The GP regression function is defined as:

$$\boldsymbol{y}_{n_v}^{(v)} = f^{(v)}(\boldsymbol{x}_{n_v}^{(v)}) + \epsilon^{(v)}, \tag{8.3}$$

where $\epsilon^{(v)} \sim \mathcal{N}(0, \sigma_v^2)$ is i.i.d. additive Gaussian noise, and the index $v \in \{s, t\}$ denotes the dependence on each domain. While in a traditional GP, all data are considered to come from the same domain, in our DA approach we focus on adapting models to new domains. The objective of a GP is to infer the latent functions $f^{(v)}$, given the training dataset $\mathcal{D}^{(v)} = \{\boldsymbol{X}^{(v)}, \boldsymbol{Y}^{(v)}\}$. By following the framework of GPs (Rasmussen and Williams, 2006), we place a prior on the functions $f^{(v)}$, so that the function values $\boldsymbol{f}_{n_v}^{(v)} = f^{(v)}(\boldsymbol{x}_{n_v}^{(v)})$ follow a Gaussian distribution $p(\boldsymbol{F}^{(v)}|\boldsymbol{X}^{(v)}) = \mathcal{N}(\boldsymbol{F}^{(v)}|\boldsymbol{0}, \boldsymbol{K}^{(v)})$. Here, $\boldsymbol{F}^{(v)} = \{\boldsymbol{f}_{n_v}^{(v)}\}_{n_v=1}^{N_v}$, and $\boldsymbol{K}^{(v)} = k^{(v)}(\boldsymbol{X}^{(v)}, \boldsymbol{X}^{(v)})$ is the kernel covariance function. We use the radial basis function (RBF) kernel, defined as:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\Big(-\frac{1}{2\ell^2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2\Big), \tag{8.4}$$

where $\{\ell, \sigma_f\}$ are the kernel hyper-parameters. The regression function is then fully defined by the set of hyper-parameters (hp) $\boldsymbol{\theta} = \{\ell, \sigma_f, \sigma_v\}$. Training of the GP consists of finding the hyper-parameters that maximize the log-marginal likelihood:

$$\begin{aligned}\log p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}^{(v)}, \boldsymbol{\theta}^{(v)}) = -\operatorname{tr}\left[(\boldsymbol{K}^{(v)} + \sigma_v^2\boldsymbol{I})^{-1}\boldsymbol{Y}^{(v)}\boldsymbol{Y}^{(v)^T}\right] \\ - \log|\boldsymbol{K}^{(v)} + \sigma_v^2\boldsymbol{I}| + \text{const.}\end{aligned} \tag{8.5}$$

Given a test input $\boldsymbol{x}_*^{(v)}$ we obtain the GP predictive distribution by conditioning on the training data $\mathcal{D}^{(v)}$ as $p(\boldsymbol{f}_*^{(v)}|\boldsymbol{x}_*^{(v)}, \mathcal{D}^{(v)}) = \mathcal{N}(\mu^{(v)}(\boldsymbol{x}_*^{(v)}), V^{(v)}(\boldsymbol{x}_*^{(v)}))$ with

$$\mu^{(v)}(\boldsymbol{x}_*^{(v)}) = \boldsymbol{k}_*^{(v)^T}(\boldsymbol{K}^{(v)} + \sigma_v^2\boldsymbol{I})^{-1}\boldsymbol{Y}^{(v)} \tag{8.6}$$

$$V^{(v)}(\boldsymbol{x}_*^{(v)}) = k_{**}^{(v)} - \boldsymbol{k}_*^{(v)^T}(\boldsymbol{K}^{(v)} + \sigma_v^2\boldsymbol{I})^{-1}\boldsymbol{k}_*^{(v)}, \tag{8.7}$$

where $\boldsymbol{k}_*^{(v)} = k^{(v)}(\boldsymbol{X}^{(v)}, \boldsymbol{x}_*^{(v)})$ and $k_{**}^{(v)} = k^{(v)}(\boldsymbol{x}_*^{(v)}, \boldsymbol{x}_*^{(v)})$. For convenience we denote $\boldsymbol{\mu}_*^{(v)} = \mu^{(v)}(\boldsymbol{x}_*^{(v)})$ and $V_{**}^{(v)} = V^{(v)}(\boldsymbol{x}_*^{(v)})$. In most applications, the GP mean function is used as the point estimate of the output targets. However, this is a generic model, i.e., it is not optimized to achieve the best performance on each target person. We describe below the adaptation approach based on GPs that we devise for personalized estimation of mood, health and stress levels for each target person.

8.6.1.2 GP Adaptation with Posterior Adaptation

The probabilistic nature of GPs allows us to easily incorporate new data of a target person into the model, without re-training the model. This results in a posterior distribution of GPs, rendering a personalized regression function specifically tuned to the target person. To this end, we exploit the Bayesian adaptation approach proposed in (Liu and Vasconcelos, 2015; Eleftheriadis et al., 2016). The GP model adaptation consists of the following three key components:

1. A GP trained on the source data with likelihood $p(\boldsymbol{Y}^{(s)}|\boldsymbol{X}^{(s)}, \boldsymbol{\theta})$ and *hp* $\boldsymbol{\theta}$ is trained as a base model, and is defined by Eqs. (8.6–8.7).
2. The posterior distribution of the base GP model is then used as a prior for the GP evaluated on target (adaptation) data $p(\boldsymbol{Y}^{(t)}|\boldsymbol{X}^{(t)}, \mathcal{D}^{(s)}, \boldsymbol{\theta})$.
3. The posterior distribution over the target data is then corrected to account for the adaptation data $\mathcal{D}^{(t)}$ of the target person.

The prior over the target data in the second step is given by applying Eqs. (8.6–8.7) on $\boldsymbol{X}^{(t)}$ as:

$$\boldsymbol{\mu}^{(t|s)} = {\boldsymbol{K}_{st}^{(s)}}^T(\boldsymbol{K}^{(s)} + \sigma_s^2\boldsymbol{I})^{-1}\boldsymbol{Y}^{(s)} \tag{8.8}$$

$$\boldsymbol{V}^{(t|s)} = \boldsymbol{K}_{tt}^{(s)} - {\boldsymbol{K}_{st}^{(s)}}^T(\boldsymbol{K}^{(s)} + \sigma_s^2\boldsymbol{I})^{-1}\boldsymbol{K}_{st}^{(s)}, \tag{8.9}$$

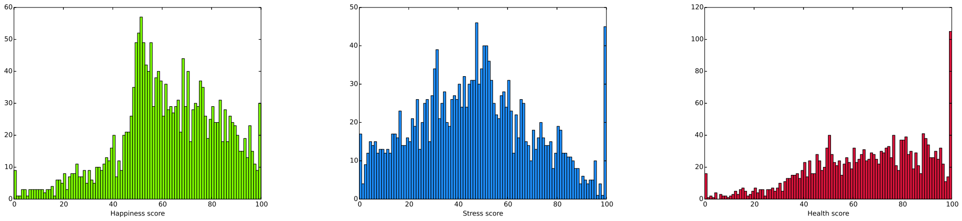
where $\boldsymbol{K}_{tt}^{(s)} = k^{(s)}(\boldsymbol{X}^{(t)}, \boldsymbol{X}^{(t)}), \boldsymbol{K}_{st}^{(s)} = k^{(s)}(\boldsymbol{X}^{(s)}, \boldsymbol{X}^{(t)})$, and the superscript $t|s$ denotes the conditioning order. Given the above prior and a test input $\boldsymbol{x}_*^{(t)}$ (i.e., features of target person), the correct form of the adapted posterior after observing the target person adaptation data is given by:

$$\mu_{ad}^{(s)}(\boldsymbol{x}_*^{(t)}) = \boldsymbol{\mu}_*^{(s)} + {\boldsymbol{V}_*^{(t|s)}}^T(\boldsymbol{V}^{(t|s)} + \sigma_s^2\boldsymbol{I})^{-1}(\boldsymbol{Y}^{(t)} - \boldsymbol{\mu}^{(t|s)}) \tag{8.10}$$

$$V_{ad}^{(s)}(\boldsymbol{x}_*^{(t)}) = V_{**}^{(s)} - {\boldsymbol{V}_*^{(t|s)}}^T(\boldsymbol{V}^{(t|s)} + \sigma_s^2\boldsymbol{I})^{-1}\boldsymbol{V}_*^{(t|s)}, \tag{8.11}$$

with $\boldsymbol{V}_*^{(t|s)} = k^{(s)}(\boldsymbol{X}^{(t)}, \boldsymbol{x}_*^{(t)}) - k^{(s)}(\boldsymbol{X}^{(s)}, \boldsymbol{X}^{(t)})^T(\boldsymbol{K}^{(s)} + \sigma_s^2\boldsymbol{I})^{-1}k^{(s)}(\boldsymbol{X}^{(s)}, \boldsymbol{x}_*^{(t)})$. Our personalized GP model based on posterior adaptation to the target person data (DA-GP) is fully defined by Eqs. (8.10–8.11). Note that the final personalized prediction is a combination of the generic (base) model based on the source data only, and a correction term, which shifts the GP mean toward the feature distribution of the target person, while reducing the model's uncertainty in the estimated output. In this way, the model automatically adapts to the range of, for instance, stress levels specific to the target person.

8.6.2 *Regression experiments*



(a) Mood (b) Stress (c) Health

Figure 8.14: Distribution of self-report labels in the data. Students frequently report feeling healthy and stressed.

In the previous two sections, we chose to discard the middle 20-40% of wellbeing labels, comprising the most neutral scores (i.e. those most near the median), in order to disambiguate the classification labels. Here, we use regression techniques to directly predict the ordinal label values that were reported, meaning we are able to make predictions for any of the available days. Figure 8.14 shows the distribution of self-report values in the data.

To ensure that we are able to train robust personalized prediction models, we restrict our attention to only those participants who provided at least 25 days of data in which all data sources are present. Since the data are noisy, this reduces the dataset to a total of 69 participants and 1895 days worth of data. These samples are then divided into non-overlapping training, validation, and testing sets using a 60/20/20% split.

The personalized models were compared to their single-task learning (STL) counterparts, a standard GP and NN. The code for the GP and NN models was written using gpml Matlab code[2] and Tensorflow, respectively. The GP model hyper-parameters were learned using conjugate-gradient optimization (Rasmussen and Williams, 2006). For the adaptation setting, we used the validation data of each target person.

In training both the STL and MTL NN, a grid search was used to

[2] http://www.gaussianprocess.org/gpml/code/

select hyperparameters (including the number and size of hidden layers) by assessing performance on the validation set. By initializing the MTL NN with the pre-trained weights of the STL NN, we found we could successfully train a much higher capacity personalized network. In the end, we found that an architecture of four hidden layers with sizes 2048, 1024, 512, and 256, a dropout factor of 0.25, and no L2 regularization gave the best performance for the STL NN, and we therefore adopted the same architecture for the MTL-NN. However, we found that while a batch size of 32 was effective for training the STL NN, a batch size of 1 gave the best results for training the MTL-NN. This could be because minibatch updates are only an unbiased estimate of the true gradient when the samples within a minibatch are uncorrelated, which necessitates a batch size of 1 in the MTL setting.

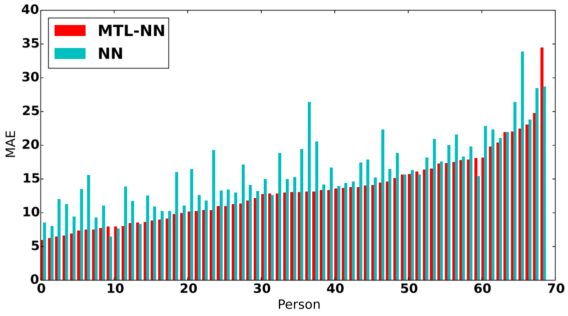
8.6.3 Regression results

	Model	**Mood**	**Stress**	**Health**	**Total**
Traditional	GP	16.0	17.2	16.7	16.6
	NN	15.0	17.1	16.5	16.2
Personalized	DA-GP	14.8	16.4	**14.6**	15.3
	MTL-NN	**13.0**	**14.1**	**12.9**	**13.3**

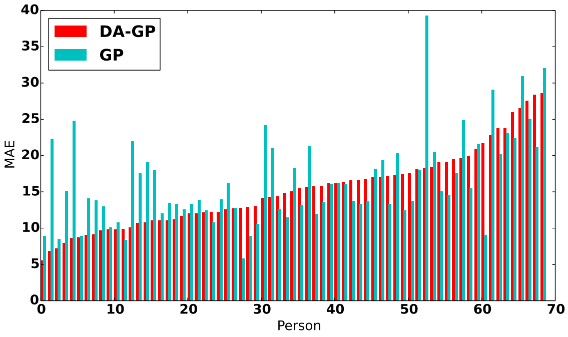
Table 8.9: MAE in predicting wellbeing on the held-out test set, for both traditional ML models and personalized models. Bolded entries represent significant improvements over the non-personalized version of the model ($p < .05$).

The Mean Absolute Error (MAE) obtained for both the traditional and personalized models is shown in Table 8.9. While both the MTL-NN and DA-GP offer reduced MAE in predicting each of the 3 wellbeing scores, paired-samples t-tests revealed that these differences are only consistently significantly different for the MTL-NN model. This difference could be due to the fact that the MTL-NN model has implicit knowledge of each of the participants' ID, due to the way the model is constructed. In contrast, the DA-GP must learn which samples are most similar to each person, which constitutes a more difficult problem. Further, the small number of samples per person in the validation set (~ 5 in this case) makes it difficult for the DA-GP to perform a robust posterior correction (Eqs.(6-7)). Note that we also attempted learning of the base GP model using both training and validation data, however, the results only marginally improved over the GP trained using training data only. This clearly shows the benefits of the proposed personalized GP adaptation scheme.

Figure 8.15 shows the MAE for each person. For 61/69 people, the personalized MTL-NN provides lower error than the generic NN.



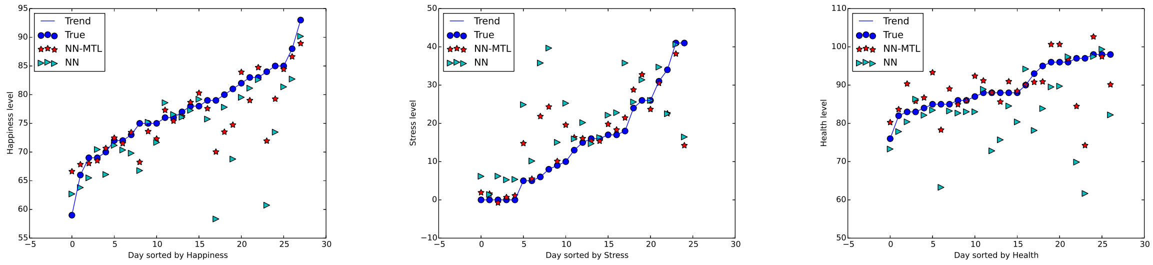
(a) NN models



(b) GP models

Figure 8.15: MAE per person for both the personalized and generic models. For 61 out of 69 participants, MAE is lower with the personalized MTL-NN model. For GPs, personalization is better for 40 out of 69 participants.

While this effect is not as strong for the DA-GP, having a personalized model still benefits the majority of people. Clearly, personalization can not only provide performance advantages across all participants, but it ensures that there are fewer people for whom the model cannot make accurate predictions.



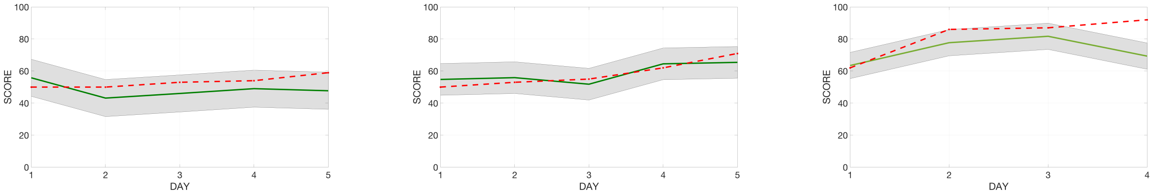
(a) Mood

(b) Stress

(c) Health

Figure 8.16: NN and MTL-NN predictions for each outcome for a randomly selected subject compared to the ground truth mood report data, which has been sorted by intensity and connected with a trend line.

Figure 8.16 shows the actual predictions of the NN and MTL-NN on each outcome label for three randomly selected participants. As is evident in the figure, the MTL-NN is able to provide a close fit to the ground truth data. Figure 8.16 also helps to demonstrate the degree of individual variability within the data; while the average Health report was 65.60 (SD=23.08), the participant in Figure 8.16 (c) only reports Health scores ranging from 76-98. This could explain why the predictions of the non-personalizd NN are so frequently drawn downwards toward the group average; it has no ability to learn to adapt its predictions to this participants' unique health pattern. Similarly, Figure 8.17 shows the predicted mean and variance of the DA-GP over the held-out test data.



(a) Mood (b) Stress (c) Health

Figure 8.17: Predicted mean and variance over the held-out test data as learned by the DA-GP. The lines in red and green represent the reported and predicted scores, respectively. As can be seen from depicted certainty levels (in gray), the predicted values are within one standard deviation (uncertainty) intervals estimated by GP.

In addition to assessing the absolute differences between the model's predictions and the ground truth, we are interested in determining if the models are able to capture the true trends underlying participants' mood and wellbeing levels; in other words, which models provide an overall better fit to the data? We assess this using the Intraclass Correlation Coefficient (ICC(3,1)), which not only measures the association between the model's predictions and the true ratings (as in Pearson's correlation), but also penalizes absolute distance from the ground truth. We find that the performance benefits offered by the personalized MTL-NN and DA-GP are substantially more pronounced in terms of ICC, as shown in Table 8.10. In this case, the benefit of the DA-GP is clearly apparent, as it provides a 100% improvement above the generic GP baseline. This is likely because the adjustment made to adapt the mean and variance of the posterior distribution for each person allows the DA-GP to adapt its predictions toward each person's true mean mood value. This dramatic improvement in ICC therefore suggests that the personalized models can more accurately distinguish the participants that have the lowest mood and health, and the highest stress - an important ability for most clinical applications.

	Model	**Mood**	**Stress**	**Health**	**Total**
Traditional	GP	.176	.358	.286	.274
	NN	.262	.422	.373	.352
Personalized	DA-GP	**.461**	**.587**	**.606**	**.551**
	MTL-NN	**.441**	**.621**	**.613**	**.558**

Table 8.10: ICC, a measure of model fit, in predicting well-being on the held-out test set. Bolded entries indicate an improvement of at least 50% over the non-personalized model.

8.6.4 Discussion

This work has empirically demonstrated that the performance of machine learning mood prediction systems can be meaningfully enhanced by personalizing those models in a principled way. We have outlined two methods for accomplishing personalization; by using a Domain Adaptation approach to adapt the posterior distribution of the model towards each person's unique mood and wellbeing level, and by using Multitask Learning to train a deep neural network with specialized final layers for each person. Not only do these models provide 13-22% lower average error than traditional models in making fine-grained predictions about participants' outcomes, but we find that the personalized models provide a significantly better fit to the data, improving ICC by as much as 160% above the generic GP baseline. These performance improvements may have important clinical benefits, such as enabling a model to better distinguish between participants who are severely unhappy or stressed, which then can enable more relevant and targeted treatments for improving wellbeing.

Clearly, personalization can provide advantages in predicting mood and wellbeing, a problem where interindividual variability is high. However, it is important to note that by their nature, these models are person-dependent; that is, they require at least some labeled data from each person in order to be trained. While the DA and MTL approaches discussed here provide the advantage that they can be trained even when there is not enough data per-person to train many individual person-specific models, in our case we still require at least 15 days of training data per person. This implies that a new user of a mood prediction system built using these models would have to input their mood and wellbeing for 15 days to obtain the level of performance presented here.

While asking users to report their mood for roughly two weeks is not unreasonable, given that most users of a quantified-self device or system continue to use it for about six months (Patel et al., 2015), it is important to note that simple extensions to the models can allow them to make predictions in the absence of any self-reports from a given user. For example, the MTL-NN could make predictions for a new user by feeding the data through each output head and averaging the predictions; we would expect this approach to recover approximately the performance of the impersonal NN model. For GPs, adapting to a new person is even simpler. In the case of the

DA-GP posterior model, with only a few days of labeled data (e.g. 5, in this case), it is possible to notably improve the predictions. It is not necessary that a person's data be part of the training set to perform the adaption. Also, multi-task extensions of GPs are another venue to pursue when personalizing models, in order to take advantages of the MTL paradigm.

8.7 Conclusions

This work has demonstrated that accounting for individual differences through Multi-task Learning and Domain Adaptation can substantially improve mood and wellbeing prediction performance. This performance enhancement is not simply due to the application of MTL, but rather through the ability of MTL to allow each person to have a model customized for them, but still benefit from the data of other people through hidden layers of a deep neural network, kernel weights, or a shared prior. The methods we have explored offer different strengths, including the ability to learn how the importance of feature types differs across tasks, and the ability to learn implicit groupings of users.

A major limitation of this research relates to the relatively small sample size. With data from more individuals and with longer monitoring per person, it may be possible to build time series models for forecasting, which could be even more accurate and powerful if personalized. In the ideal case, we would like to be able to predict a person's wellbeing far into the future, rather than just one day in advance. Given the scope of the work undertaken here, there are many other aspects that could be enhanced. For example, the features were selected using a generic method that did not take the classifier type into consideration. We also do not use several features suggested by other work as important for mental health and wellbeing (Sano, 2015). Future work will more deeply examine these cases and their interplay with a personal model. Further, we can also incorporate exciting new techniques for improving MTL in future, for example by combining the hierarchical Bayesian approach with deep neural networks (DNNs) (Salakhutdinov et al., 2013), or training DNNs that are able to quickly adapt parameters in order to learn new tasks (Finn et al., 2017a). Since the publication of this work, our group has continued research into forecasting mood, stress, and health, and has found that using recurrent models such as LSTMs to learn trends in behavior can lead to further improvements in forecasting performance (Umematsu et al., 2019).

It is worth emphasizing that the presented algorithms could provide the ability to make predictions about a novel person who has not provided any self-report labels. If this person is willing to complete a personality inventory, predictions can be made immediately using the MTL-NN and MTMKL models, which are based on K-means clusters computed from personality and gender data. The HBLR model can be extended to make mood predictions for a novel user who has not provided classification labels, by applying MCMC to her data (Xue et al., 2007). In future work, we will assess the classification accuracy of these models on novel participants. A strongly motivating application goal is to be able to detect individuals with low wellbeing or mental health in order to guide prevention or early intervention efforts. If new users were only required to install an app that did not depend on them inputting mood data, interventions would be able to reach a larger population.

Another aim of this research is to generate hypotheses about problematic behaviors that are indicative of low mood and mental health, a line of research that our group has continued to investigate (Nosakhare and Picard, 2019). Through examination of the model weights and clusters, we hope to gain insight into the behaviors that are significant wellbeing predictors for people with different personalities and lifestyles. Once hypotheses related to these behaviors have been refined, we can test them via causal inference techniques such as counterfactual reasoning. These inferences would be useful for anyone wishing to know what types of behaviors best promote a happy, calm, and healthy state.

Finally, we hope that by providing the code for these techniques, other authors will be encouraged to use them to personalize models for a wide variety of problems in which interindividual variability is important. When used in conjunction with the analysis techniques outlined here, these models may not only lead to the discovery of interesting insights across many problems, but may help to significantly enhance performance in predicting difficult, ambiguous, and personal outcomes.

8.8 Statement of contributions

This chapter comprises material from 4-5 papers. The initial experiments described in Section 8.4 began as a class project which myself, Sara Taylor, Asaf Azaria, and Asma Ghandeharioun worked on jointly, in consultation with Akane Sano and Rosalind Picard. Asaf wrote the code for the GMM location model, while Asma, Sara and I wrote code to extract physiology, smartphone, and survey features. I wrote the code for the machine learning models and ran those experiments. Later, Sara and I continued developing the data processing code, and I wrote an algorithm to extract steps from the accelerometer data and proposed accelerometer- and temperature-weighted EDA features, while Sara wrote code to scrape weather data and extract features from it. Sara also used her knowledge of signal processing to contribute significantly to the development of the physiology features. I proposed and conducted the feature evaluation.

The idea for using multi-task learning to personalize machine learning models to the individual was something I pushed forward after my Master's thesis advisor, Cristina Conati, encouraged me to read Melih Kandemir's PhD thesis. I proposed adopting Kandemir's MTMKL model to the SNAPSHOT data. Sara and I worked together on this idea, jointly coding MTMKL and using it to multi-task over related wellbeing labels, leading to an initial workshop publication. We then continued in this direction by actually using MTMKL for personalization, and adding the personalized neural network and HBLR models. I was responsible for the neural network code. Sara and I worked together to code the HBLR model given the (often erroneous) descriptions in the original paper; Sara took a lead role in translating these equations into code. During this phase of the project, we also worked with Ehi Nosakhare, who conducted a wide literature review and consulted with us on many aspects of the project, as well as Akane Sano who continued to provide support and advice. Rosalind Picard advised throughout.

The personalized regression experiments were conducted in service of a fourth paper which I wrote with Ognjen Rudovic and Rosalind. I wrote code for and trained the neural network models, and Ognjen wrote code for and trained the Gaussian Process models. I devised and conducted the analysis of the results.

9 *Conclusion*

Social learning is a core component of humans' impressive intelligence, enabling us to rapidly transmit knowledge and adapt to new situations. The work presented in this dissertation developed techniques for improving the social learning capabilities of artificial intelligence (AI), using and improving upon the tools of machine learning, deep learning, and reinforcement learning.

Chapter 3 introduced novel mechanisms for social learning from other AI agents in a multi-agent system. Agents modeled the effect of their actions on the behavior of other agents, allowing them to coordinate and communicate more effectively. We note that multi-agent training provides the advantage of allowing for social learning in simulation, where it is possible to learn complex coordinated behaviors by scaling up to millions of interactions.

However, learning from other AI agents will be insufficient to solve many real-world tasks where humans are the ultimate authority on good performance, such as conversation (Chapter 4), music generation (Section 4.3), or drawings and art (Chapter 5). Yet asking humans to manually label correct performance does not scale effectively, as shown in Chapter 4. Therefore, in Chapters 4 and 5, I showed that ML models can enhance their performance by learning from implicit cues gleaned from interacting with humans, allowing them to learn from human feedback without additional human effort. These models were intrinsically motivated to produce positive reactions in humans (such as smiles).

Eventually, we would like to build an AI agent that is intrinsically motivated to increase human wellbeing, satisfaction, and flourishing, because such an agent is more likely to be safe and useful. Doing so may require going beyond simple cues like facial expressions, instead using ML to infer people's underlying mental and affective states. Therefore, Chapter 6 presented techniques for detecting when humans are enjoying a conversation and experiencing bonding and rapport, and Chapter 8 developed methods for predicting an individual's stress, health, and happiness using personalized multi-task learning.

Learning from humans presents several challenges. Data are often limited, noisy, and even missing. Chapter 7 presents several techniques for working with noisy data and inferring missing elements. When training a system meant for human use, it may not be possible to allow the system to learn online during interactions with humans, because it could potentially learn unsafe or inappropriate behavior. Chapter 4 presented novel techniques for learning from a fixed batch of limited human interaction data using RL, even when it is not possible to learn online and explore. Finally, many of the techniques presented in this dissertation leverage transfer learning to first train a reasonable policy, and then fine-tune with limited human feedback. This ensures that scarce human feedback data are used as effectively and efficiently as possible.

9.1 *Future work*

This dissertation lays the groundwork for creating a unified pipeline for training agents with enough social intelligence to flexibly coordinate with and learn from humans in the real world. Training from scratch with purely human data is unlikely to scale effectively. Therefore, we can use multi-agent training in simulation as a first step. We can create new social learning mechanisms inspired by human social intelligence, and test them in simulation. Once agents have learned reasonable policies for inferring the intentions of other agents and coordinating with them, we can use transfer learning to deploy these agents to interact with humans. By sensing human feedback, both implicitly and potentially through better language understanding, these models can continue to learn online in the real-world. Below, I describe the steps of this research agenda in detail.

The area of multi-agent social learning has only begun to be explored. An abundance of insights from social and cognitive psychology (e.g. Tomasello (2009a); Laland (2017); Barkow et al. (1975); Leibo et al. (2019); Humphrey (1976); Rendell et al. (2010)) can be leveraged to create AI agents which are able to coordinate with other agents, socially transmit information, and adapt more flexibly and intelligently to new environments. I have shown in this dissertation that counterfactual reasoning and causal inference, when combined with modeling the behavior of other agents, is a promising direction for improving multi-agent cooperation. Counterfactual reasoning can be used to allow an agent to understand the causal impact of its actions on another agent. In future work, we can use similar techniques to allow an agent to swap itself into the place of another agent, and compute whether an action it would like to take would affect the other agent positively or negatively. Essentially, we allow the agent to ask, "How would I feel if the other agent did this to me?". By rewarding the agent for only taking actions which it predicts will help others, we can create a *golden rule* incentive that can drive cooperation. These cooperative approaches could be scaled to domains that involve complex, dynamic, ad-hoc coordination between multiple agents with different intentions.

Beyond causal reasoning, there are many ways to think about social incentives that can improve exploration and generalization. Multi-agent learning leads to a combinatorial explosion in the size of the action space, making exploration extremely difficult. This could potentially be alleviated by rewarding specialization among the agents through a multi-agent version of the diversity incentive proposed by Eysenbach et al. (2018).

Multi-agent communication could be improved through a motivation to create messages that are attended to by multiple other agents. Interestingly, the motivation to garner attention from others appears to exist in both humans and primates (Barkow et al., 1975). Further, if we assume that agents are building models of other agents, we can give agents a motivation to be positively modeled by others, akin to impression-management strategies in humans. This could also help drive cooperative, prosocial behavior. Finally, curiosity over other agents could drive more sophisticated interactions with, and modeling of, others.

We can use human coordination games as an exciting test-bed for whether learned multi-agent policies generalize to coordinating with humans – an ability required of future AI systems deployed to the human environment. Using transfer learning techniques similar to those presented in this dissertation, agent-agent cooperative policies learned in simulation can be deployed to interact with humans in these games. Tools such as meta-learning could be leveraged to ensure that agents are able to quickly adapt to new agents which they have never encountered before. The previously developed social learning techniques can guide agents to inferring the intentions of humans in these games and learning to coordinate with them.

Finally, we would like to learn from other forms of human feedback, especially when deploying agents to the real world where rich non-verbal social cues may provide additional information about humans' internal affective states. While facial expressions and text are just two modalities that humans use to express themselves, recent research has shown that information indicative of a person's mental state can be automatically detected in their tone of voice (e.g. Scherer et al. (2015)), or their body language (e.g. Jaques et al. (2016b)). Incorporating those predictions as additional feedback for training AI agents, including physical agents such as robots, represents an important next step.

If the efforts above are successful, it could potentially lead to the development of not only smarter, more functional AI, but systems that can flexibly adapt to and coordinate with humans, in order to optimize for our goals and preferences. Many resarchers have argued the importance of social learning for human cognitive and cultural devleopment (Harari, 2014; Laland, 2017; Henrich, 2015; Kleiman-Weiner, 2018; Herrmann et al., 2007; van Schaik and Burkart, 2011). I believe that improving the social intelligence of machine learning algorithms could be the impetus for a significant breakthrough in AI, just as improvements in social intelligence were critical for human development.

Bibliography

Abadi, M. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM.

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*.

Adedokun, O. A. and Burgess, W. D. (2011). Analysis of paired dichotomous data: A gentle introduction to the mcnemar test in spss. *Journal of MultiDisciplinary Evaluation*, 8(17):125–131.

Agarwal, R., Schuurmans, D., and Norouzi, M. (2019). Striving for simplicity in off-policy deep reinforcement learning. *arXiv preprint arXiv:1907.04543*.

Agrawal, P. (2018). *Computational sensorimotor learning*. PhD thesis, University of California, Berkeley.

Aichele, S., Rabbitt, P., and Ghisletta, P. (2016). Think fast, feel fine, live long: A 29-year study of cognition, health, and survival in middle-aged and older adults. *Psychological science*, 27(4):518–529.

Ambady, N. and Rosenthal, R. (1992). Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological bulletin*, 111(2):256.

Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., et al. (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature medicine*, page 1.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009).

A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

Argyle, M. (1972). Non-verbal communication in human social interaction.

Arora, S., Cohen, N., and Hazan, E. (2018). On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*.

Arroyo, I. et al. (2009). Emotion sensors go to school. In *AIED*, volume 200, pages 17–24.

Astrid, M. et al. (2010). How our personality shapes our interactions with virtual characters-implications for research and development. In *IVA*, pages 208–221.

Avola, D. et al. (2013). Human body language analysis: A preliminary study based on kinect skeleton tracking. In *ICIAP*, pages 465–473.

Azizzadenesheli, K., Brunskill, E., and Anandkumar, A. (2018). Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE.

Bahdanau, D., Brakel, P., Xu, K., Goyal, A., Lowe, R., Pineau, J., Courville, A., and Bengio, Y. (2016). An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. (2019). Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*.

Bandura, A. and Walters, R. H. (1977). Social learning theory.

Barkow, J. H., Akiwowo, A. A., Barua, T. K., Chance, M., Chapple, E. D., Chattopadhyay, G. P., Freedman, D. G., Geddes, W., Goswami, B., Isichei, P., et al. (1975). Prestige and culture: a biosocial interpretation [and comments and replies]. *Current Anthropology*, 16(4):553–572.

Barreira, P., Basilico, M., and Bolotnyy, V. (2018). Graduate student mental health: Lessons from american economics departments. Technical report, Working paper). Retrieved November 29, 2018, from https://scholar. harvard âĂę.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945.

Barton, S. L., Waytowich, N. R., Zaroukian, E., and Asher, D. E.

(2018). Measuring collaborative emergent behavior in multi-agent reinforcement learning. *arXiv preprint arXiv:1807.08663*.

Bauer, G. and Lukowicz, P. (2012). Can smartphones detect stress-related changes in the behaviour of individuals? In *Int. Conf. on Pervasive Comput. and Commun.*, pages 423–426. IEEE.

Baxter, J. (1997). A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1):7–39.

Bellman, R. and Kalaba, R. (1957). Dynamic programming and statistical communication theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43(8):749.

Bengio, Y. (2014). Evolving culture versus local minima. In *Growing Adaptive Machines*, pages 109–138. Springer.

Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Bhatt, A., Argus, M., Amiranashvili, A., and Brox, T. (2019). Cross-norm: Normalization for off-policy td reinforcement learning. *arXiv preprint arXiv:1902.05605*.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. (2012). Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bodie, G. D., St. Cyr, K., Pence, M., Rold, M., and Honeycutt, J. (2012). Listening competence in initial interactions i: Distinguishing between what listening is and what listeners do. *International Journal of Listening*, 26(1):1–28.

Bogin, B., Geva, M., and Berant, J. (2018). Emergence of communication in an interactive world with consistent speakers. *arXiv preprint arXiv:1809.00549*.

Bogomolov, A. et al. (2013). Happiness recognition from mobile phone data. In *Social Computing (SocialCom), 2013 International Conference on*, pages 790–795. IEEE.

Bogomolov, A., Lepri, B., Ferron, M., Pianesi, F., and Pentland, A. (2014). Daily stress recognition from mobile phone data, weather conditions and individual traits. In *ICME*, pages 477–486. ACM.

Bottou, L. (1998). Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142.

Boucsein, W. (2012). *Electrodermal activity*. Springer Science+Business Media, LLC.

Bower, B. et al. (2010). Poor reported sleep quality predicts low positive affect in daily life among healthy and mood-disordered persons. *Journal of sleep research*, 19(2):323–332.

Breazeal, C. et al. (1998). A motivational system for regulating human-robot interaction. In *Aaai/iaai*, pages 54–61.

Breazeal, C., Kidd, C. D., Thomaz, A. L., Hoffman, G., and Berlin, M. (2005). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ international conference on intelligent robots and systems*, pages 708–713. IEEE.

Breazeal, C. L. (2000). *Sociable machines: Expressive social exchange between humans and robots*. PhD thesis, Massachusetts Institute of Technology.

Brebner, J. (1990). Personality factors in stress and anxiety. *Cross-cultural anxiety*, 4:11–19.

Brochu, E. (2010). *Interactive Bayesian optimization: learning user preferences for graphics and animation*. PhD thesis, University of British Columbia.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym.

Butler, E. (2011). Temporal interpersonal emotion systems the âĂIJtiesâĂİ that form relationships. *Pers. and Soc. Psych. Review*, 15(4):367–393.

Caliskan-Islam, A., Bryson, J. J., and Narayanan, A. (2016). Semantics derived automatically from language corpora necessarily contain human biases. *arXiv preprint arXiv:1608.07187*.

Canzian, L. and Musolesi, M. (2015). Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 1293–1304. ACM.

Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., and Clark, S. (2018). Emergent communication through negotiation. *arXiv preprint arXiv:1804.03980*.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings* *of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299.

Capdepuy, P., Polani, D., and Nehaniv, C. L. (2007). Maximization of potential information flow as a universal utility for collective behaviour. In *Artificial Life, 2007. ALIFE'07. IEEE Symposium on*, pages 207–213. Ieee.

Carneiro, D., Novais, P., Augusto, J. C., and Payne, N. (2017). New methods for stress assessment and monitoring at the workplace. *IEEE Transactions on Affective Computing*.

Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. (2016). Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742.

Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

Carver, C. S. and Scheier, M. F. (2011). *Perspectives on personality*. Pearson Higher Ed.

Cheng, H. and Furnham, A. (2003). Personality, self-esteem, and demographic predictions of happiness and depression. *Personality and individual differences*, 34(6):921–942.

Chentanez, N., Barto, A. G., and Singh, S. P. (2005). Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288.

Chernova, S. and Thomaz, A. L. (2014). Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121.

Cho, J. and Kato, S. (2011). Detecting emotion from voice using selective bayesian pairwise classifiers. In *2011 IEEE Symposium on Computers & Informatics*, pages 90–95. IEEE.

Choi, E., Lazaridou, A., and de Freitas, N. (2018). Compositional obverter communication learning from raw visual input. *arXiv preprint arXiv:1804.02341*.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307.

Clark, L., Watson, D., and Mineka, S. (1994). Temperament, personality, and the mood and anxiety disorders. *Journal of abnormal psychology*, 103(1):103.

Cohen, S. et al. (1991). Psychological stress and susceptibility to the common cold. *New England journal of medicine*, 325(9):606–612.

Cohen, S. and Herbert, T. (1996). Health psychology: Psychological factors and physical disease from the perspective of human psychoneuroimmunology. *Annu. rev. of psychology*, 47(1):113–142.

Cohen, S. and Wills, T. (1985). Stress, social support, and the buffering hypothesis. *Psychological bulletin*, 98(2):310.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Crandall, J. W., Oudah, M., Chenlinangjia, T., Ishowo-Oloko, F., Abdallah, S., Bonnefon, J., Cebrián, M., Shariff, A., Goodrich, M. A., and Rahwan, I. (2017). Cooperating with machines. *CoRR*, abs/1703.06207.

Crawford, V. P. and Sobel, J. (1982). Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pages 1431–1451.

Cuperman and Ickes (2009). Big five predictors of behavior and perceptions in initial dyadic interactions. *J. of Pers. and Soc. Psych.*, 97(4):667.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Czeisler, C. et al. (1986). Bright light resets the human circadian pacemaker independent of the timing of the sleep-wake cycle. *Science*, 233(4764):667–671.

Damasio, A. R. (1994). Descartes' error: Emotion, rationality and the human brain.

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics.

Degris, T., White, M., and Sutton, R. S. (2012). Off-policy actor-critic. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 179–186. Omnipress.

Deng, J., Xu, X., Zhang, Z., Frühholz, S., and Schuller, B. (2017).

Universum autoencoder-based domain adaptation for speech emotion recognition. *IEEE Signal Processing Letters*, 24(4):500–504.

Deng, J., Zhang, Z., Marchi, E., and Schuller, B. (2013). Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 511–516. IEEE.

Devlin, S., Yliniemi, L., Kudenko, D., and Tumer, K. (2014). Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 165–172. International Foundation for Autonomous Agents and Multiagent Systems.

Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., et al. (2019). The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Dinh, L. (2018). *Reparameterization in Deep Learning*. PhD thesis, Mila, Université de Montréal.

D'Mello, S. K. and Graesser, A. (2010). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-Adapted Interaction*, 20(2):147–187.

Doberenz, S. et al. (2011). Methodological considerations in ambulatory skin conductance monitoring. *Int. J. of Psychophysiology*, 80(2):87–95.

Dong, W. et al. (2011). Modeling the co-evolution of behaviors and social relationships using mobile phone data. In *Int. Conf. on Mobile and Ubiquitous Multimedia*, pages 134–143. ACM.

Duan, Y., Andrychowicz, M., Stadie, B., Ho, O. J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. (2017). One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098.

Ekman, P. and Friesen, W. (1977). Facial action coding system.

Eleftheriadis, S., Rudovic, O., Deisenroth, M., and Pantic, M. (2016). Gaussian process domain experts for model adaptation in facial behavior analysis. In *CVPR'W*, pages 18–26.

Engel, J., Hoffman, M., and Roberts, A. (2017). Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*.

Ertl, P. and Schuffenhauer, A. (2009). Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8.

Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Zidek, A., Nelson, A., Bridgland, A., Penedones, H., et al. (2018). De novo structure prediction with deeplearning based scoring. *Annu Rev Biochem*, 77:363–382.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. (2018). More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1446–1455.

Fatemi, M., El Asri, L., Schulz, H., He, J., and Suleman, K. (2016). Policy networks with two-stage training for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 101–110.

Fedor, S. and Picard, R. (2014). Ambulatory eda: Comparisons of bilateral forearm and calf locations. 51:S76–S76.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *2017 Conference on Empirical Methods in Natural Language ProcessingConference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ferguson, H. J., Scheepers, C., and Sanford, A. J. (2010). Expectations in counterfactual and theory of mind reasoning. *Language and Cognitive Processes*, 25(3):297–346.

Finn, C. (2018). *Learning to Learn with Gradients*. PhD thesis, UC Berkeley.

Finn, C., Abbeel, P., and Levine, S. (2017a). Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400.

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. (2017b). One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Fleiss, J. L., Cohen, J., and Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5):323.

Foerster, J., Assael, I. A., de Freitas, N., and Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145.

Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. (2018). Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130. International Foundation for Autonomous Agents and Multiagent Systems.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. (2017). Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*.

Forestier, S. and Oudeyer, P.-Y. (2017). A unified model of speech and tool use early development. In *39th Annual Conference of the Cognitive Science Society (CogSci 2017)*.

Fox, R., Pakman, A., and Tishby, N. (2016). Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 202–211. AUAI Press.

Fujimoto, S., Hoof, H., and Meger, D. (2018a). Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1582–1591.

Fujimoto, S., Meger, D., and Precup, D. (2018b). Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900*.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202.

Gadanho, S. C. and Hallam, J. (2001). Robot learning driven by emotions. *Adaptive Behavior*, 9(1):42–64.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, pages 8789–8798.

Gašić, M., Jurčíček, F., Thomson, B., Yu, K., and Young, S. (2011). On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 312–317. IEEE.

Gauldin (1995). *A practical approach to eighteenth-century counterpoint*. Waveland Pr Inc.

Gelada, C. and Bellemare, M. G. (2019). Off-policy deep reinforcement learning by bootstrapping the covariate shift. *arXiv preprint arXiv:1901.09455*.

Gelman, A. and Hill, J. (2007). Missing-data imputation. *Behavior research methods*, 43(2):310–30.

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Ghandeharioun, A., Shen, J., Jaques, N., Ferguson, C., Jones, N., Lapedriza, A., and Picard, R. (2019). Approximating interactive human evaluation with self-play for open-domain dialog systems. *arXiv preprint arXiv:1906.09308*.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Goldberg, L. R. (1992). The development of markers for the big-five factor structure. *Psychological assessment*, 4(1):26.

Gómez-Bombarelli, R., Duvenaud, D., Hernández-Lobato, J. M., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2016). Automatic chemical design using a data-driven continuous representation of molecules. *arXiv preprint arXiv:1610.02415*.

Gonzales, A. L., Hancock, J. T., and Pennebaker, J. W. (2010). Language style matching as a predictor of social dynamics in small groups. *Communication Research*, 37(1):3–19.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT Press.

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 *AAAI*, pages 3951–3957.

Gratch, J. et al. (2007). Creating rapport with virtual agents. In *Intelligent Virtual Agents*, pages 125–138.

Gratch, J., Okhmatovskaia, A., Lamothe, F., Marsella, S., Morales, M., van der Werf, R. J., and Morency, L.-P. (2006). Virtual rapport. In *Intelligent virtual agents*, pages 14–27. Springer.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Grünerbl, A., Muaremi, A., Osmani, V., Bahle, G., Oehler, S., Tröster, G., Mayora, O., Haring, C., and Lukowicz, P. (2015). Smartphone-based recognition of states and state changes in bipolar disorder patients. *IEEE Journal of Biomedical and Health Informatics*, 19(1):140–148.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. (2016). Continuous Deep Q-Learning with model-based acceleration. In *ICML*.

Guckelsberger, C., Salge, C., and Togelius, J. (2018). New and surprising ways to be mean. adversarial npcs with coupled empowerment minimisation. *arXiv preprint arXiv:1806.01387*.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The J. of Mach. Learning Research*, 3:1157–1182.

Ha, D. and Eck, D. (2017). A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361. JMLR. org.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865.

Hall, M. A. (1998). *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand.

Hancock, B., Bordes, A., Mazare, P.-E., and Weston, J. (2019). Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*.

Harari, Y. N. (2014). *Sapiens: A brief history of humankind*. Random House.

Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., and Malik, J. (2011). Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE.

Hashimoto, C. and Sassano, M. (2018). Detecting absurd conversations from intelligent assistant logs by exploiting user feedback utterances. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 147–156. International World Wide Web Conferences Steering Committee.

Hashimoto, T. B., Zhang, H., and Liang, P. (2019). Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*.

Hawkins, J. and Blakeslee, S. (2004). On intelligence. *New York St. Martins Griffin*, pages 156–8.

Hawkley, L. and Cacioppo, J. (2010). Loneliness matters: a theoretical and empirical review of consequences and mechanisms. *Ann. of Behavioral Medicine*, 40(2):218–227.

Hay, J. (2000). Functions of humor in the conversations of men and women. *Journal of pragmatics*, 32(6):709–742.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Healey, J. and Picard, R. (1998). Digital processing of affective signals. In *Int. Conf. on Acoustics, Speech and Signal Processing*, volume 6, pages 3749–3752. IEEE.

Hedman, E. B. (2010). *In-situ measurement of Electrodermal Activity during Occupational Therapy*. PhD thesis, MIT.

Henrich, J. (2015). *The Secret of Our Success: How culture is driving* *human evolution, domesticating our species, and making us smart*. Princeton University Press, Princeton, NJ.

Hernandez, J., Morris, R. R., and Picard, R. W. (2011). Call center stress recognition with person-specific models. In *ACII*, pages 125–134. Springer.

Herrmann, E., Call, J., Hernàndez-Lloreda, M. V., Hare, B., and Tomasello, M. (2007). Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis. *Science*, 317(5843):1360–1366.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hollerman, J. R. and Schultz, W. (1998). Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature neuroscience*, 1(4):304.

Hoque, M. and Picard, R. W. (2011). Acted vs. natural frustration and delight: Many people smile in natural frustration. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 354–359. IEEE.

Hornik, K., Stinchcombe, M., and White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560.

Horton, H. (2016). Microsoft deletes 'teen girl' ai after it became a hitler-loving sex robot within 24 hours. In *Telegraph UK*.

Horvath, A. and Greenberg, L. (1989). Development and validation of the working alliance inventory. *Journal of counseling psychology*, 36(2):223.

House, J. et al. (1988). Social relationships and health. *Science*, 241(4865):540–545.

Huang, C., Zaiane, O., Trabelsi, A., and Dziri, N. (2018a). Automatic dialogue generation with expressed emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 49–54.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck,

D. (2018b). Music transformer: Generating music with long-term structure.

Hughes, E., Leibo, J. Z., Phillips, M. G., Tuyls, K., Duéñez-Guzmán, E. A., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K. R., Koster, R., et al. (2018). Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in neural information processing systems (NIPS)*, Montreal, Canada.

Humphrey, N. K. (1976). The social function of intellect. In *Growing points in ethology*, pages 303–317. Cambridge University Press.

Hussain, M. et al. (2011). Affect detection from multichannel physiology during learning sessions with autotutor. In *AIED*, pages 131–138. Springer.

Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., et al. (2015). An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*.

Hyung-il, A. and Picard, R. (2006). Affective-cognitive learning and decision-making: The role of emotions (pdf). In *The 18th European Meeting on Cybernetics and Systems Research (EMCSR 2006)*.

Ireland, M. E., Slatcher, R. B., Eastwick, P. W., Scissors, L. E., Finkel, E. J., and Pennebaker, J. W. (2011). Language style matching predicts relationship initiation and stability. *Psychological science*, 22(1):39–44.

Jaques, N. (2014). *Predicting affect in an Intelligent Tutoring System*. PhD thesis, University of British Columbia.

Jaques, N., Ghandeharioun, A., Shen, J., Ferguson, C., Jones, N., Lapedriza, A., Gu, S., and Picard, R. (2019a). Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

Jaques, N., Gu, S., Bahdanau, D., Hernández-Lobato, J. M., Turner, R. E., and Eck, D. (2017a). Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1645–1654. JMLR. org.

Jaques, N., Kim, Y. L., and Picard, R. (2016a). Personality, attitudes, and bonding in conversations. In *International Conference on Intelligent Virtual Agents*, pages 378–382. Springer.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z., and De Freitas, N. (2019b). Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049.

Jaques, N., McCleary, J., Engel, J., Ha, D., Bertsch, F., Picard, R., and Eck, D. (2018). Learning via social awareness: Improving a deep generative sketching model with facial feedback. *Proceedings of Machine Learning Research*.

Jaques, N., McDuff, D., Kim, Y. L., and Picard, R. (2016b). Understanding and predicting bonding in conversations using thin slices of facial expressions and body language. In *International Conference on Intelligent Virtual Agents*, pages 64–74. Springer.

Jaques, N., Taylor, S., Azaria, A., Ghandeharioun, A., Sano, A., and Picard, R. (2015a). Predicting students' happiness from physiology, phone, mobility, and behavioral data. In *Affective computing and intelligent interaction (ACII), 2015 international conference on*, pages 222–228. IEEE.

Jaques, N., Taylor, S., Nosakhare, E., Sano, A., and Picard, R. (2016c). Multi-task learning for predicting health, stress, and happiness. In *NIPS Workshop on Machine Learning for Healthcare, Barcelona, Spain*.

Jaques, N., Taylor, S., Sano, A., and Picard, R. (2015b). Multi-task, multi-kernel learning for estimating individual wellbeing. In *Proc. NIPS Workshop on Multimodal Machine Learning, Montreal, Quebec*, volume 898.

Jaques, N., Taylor, S., Sano, A., and Picard, R. (2017b). Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *Proc. International Conference on Affective Computing and Intelligent Interaction (ACII), San Antonio, Texas*.

Jaques, N., Taylor, S., Sano, A., Picard, R., et al. (2017c). Predicting tomorrow's mood, health, and stress level using personalized multitask learning and domain adaptation. In *IJCAI 2017 Workshop on Artificial Intelligence in Affective Computing*, pages 17–33.

Jiang, N. and Li, L. (2016). Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661.

Jin, F. and Sun, S. (2008). Neural network multitask learning for traffic flow forecasting. In *IEEE Int'l Joint Conf. on Neural Networks*, pages 1897–1901. IEEE.

John, O. P. and Srivastava, S. (1999). The big five trait taxonomy:

History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research*, 2(1999):102–138.

Kahl, S. and Kopp, S. (2015). Modeling a social brain for interactive agents: integrating mirroring and mentalizing. In *IVA*, pages 77–86.

Kahn, G., Villaflor, A., Pong, V., Abbeel, P., and Levine, S. (2017). Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*.

Kakade, S. M. (2002). A natural policy gradient. In *Advances in neural information processing systems (NIPS)*, volume 14, pages 1531–1538.

Kandemir, M. et al. (2014). Multi-task and multi-view learning of user state. *Neurocomputing*, 139:97–106.

Kapoor, A. and Picard, R. W. (2005). Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 677–682. ACM.

Kappeler-Setz, C. et al. (2013). Towards long term monitoring of electrodermal activity in daily life. *Pers. ubiquit. comput.*, 17(2):261–271.

Kappen, H. J., Gómez, V., and Opper, M. (2012). Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182.

Karpathy, A. (2016). Deep reinforcement learning: Pong from pixels.

Kawaguchi, K. (2016). Deep learning without poor local minima. In *Advances in neural information processing systems*, pages 586–594.

Keller, A. et al. (2012). Does the perception that stress affects health matter? the association with health and mortality. *Health Psychology*, 31(5):677.

Kim, B. and Pineau, J. (2016). Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, 8(1):51–66.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kisch, J. et al. (2005). Aspects of suicidal behavior, depression, and treatment in college students: results from the spring 2000 national college health assessment survey. *Suicide and Life-Threatening Behavior*, 35(1):3–13.

Kleiman-Weiner, M. (2018). *Computational foundations of human social intelligence*. PhD thesis, Massachusetts Institute of Technology.

Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., and Tenenbaum, J. B. (2016). Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*.

Klerman, G. and Weissman, M. (1989). Increasing rates of depression. *JAMA*, 261(15):2229–2235.

Klimstra, T. et al. (2011). Come rain or come shine: individual differences in how weather affects mood. *Emotion*, 11(6):1495.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: A universal agent-centric measure of control. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 128–135. IEEE.

Knox, W. B. and Stone, P. (2009). Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM.

Kocielnik, R. et al. (2013). Smart technologies for long-term stress monitoring at work. In *Comput.-Based Medical Syst.*, pages 53–58. IEEE.

Koelstra, S. et al. (2012). Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31.

Kohler, C. et al. (2004). Differences in facial expressions of four universal emotions. *Psychiatry research*, 128(3):235–244.

Koldijk, S., Neerincx, M. A., and Kraaij, W. (2017). Detecting work stress in offices by combining unobtrusive sensors. *IEEE Transactions on Affective Computing*, PP(99):1–1.

Kort, B., Reilly, R., and Picard, R. W. (2001). An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Proceedings IEEE International Conference on Advanced Learning Technologies*, pages 43–46. IEEE.

Kosti, R., Alvarez, J., Recasens, A., and Lapedriza, A. (2019). Context based emotion recognition using emotic dataset. *IEEE transactions on pattern analysis and machine intelligence*.

Krening, S., Harrison, B., Feigh, K. M., Isbell, C. L., Riedl, M., and Thomaz, A. (2017). Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kujawa, A., Dougherty, L., Durbin, C. E., Laptook, R., Torpey, D., and Klein, D. N. (2014). Emotion recognition in preschool children: Associations with maternal depression and early parenting. *Development and psychopathology*, 26(1):159–170.

Kumar, A., Fu, J., Tucker, G., and Levine, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*.

Kuzovkin, I., Vicente, R., Petton, M., Lachaux, J.-P., Baciu, M., Kahane, P., Rheims, S., Vidal, J. R., and Aru, J. (2018). Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Communications biology*, 1(1):107.

Laland, K. N. (2017). *Darwin's unfinished symphony : how culture made the human mind / Kevin N. Laland*. Princeton University Press Princeton.

Lane, N. et al. (2010). A survey of mobile phone sensing. *Commun. Magazine, IEEE*, 48(9):140–150.

Lazaridou, A., Hermann, K. M., Tuyls, K., and Clark, S. (2018). Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Leibo, J. Z., Hughes, E., Lanctot, M., and Graepel, T. (2019). Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742*.

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. (2017). Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents* *and MultiAgent Systems*, pages 464–473. International Foundation for Autonomous Agents and Multiagent Systems.

Lenat, D. B. (1976). Am: An artificial intelligence approach to discovery in mathematics as heuristic search. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018a). Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., and Dolan, B. (2016a). A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 994–1003.

Li, J. and Jurafsky, D. (2017). Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209.

Li, J., Miller, A. H., Chopra, S., Ranzato, M., and Weston, J. (2016b). Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016c). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Li, J., Wang, X., and Hovy, E. (2014). What a nasty day: Exploring mood-weather relationship from twitter. In *Int'l Conf. on Info. and Knowledge Management*, pages 1309–1318. ACM.

Li, Z., Kiseleva, J., and de Rijke, M. (2018b). Dialogue generation: From imitation learning to inverse reinforcement learning. *arXiv preprint arXiv:1812.03509*.

LiKamWa, R. et al. (2013). Moodscope: building a mood sensor from smartphone usage patterns. In *Int. Conf. on Mobile systems, applications, and services*, pages 389–402. ACM.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Liu, B. and Lane, I. (2017). Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.

Liu, B., Singh, S., Lewis, R. L., and Qin, S. (2014). Optimal rewards for cooperative agents. *IEEE Transactions on Autonomous Mental Development*, 6(4):286–297.

Liu, B., Tür, G., Hakkani-Tür, D., Shah, P., and Heck, L. (2018). Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069.

Liu, B. and Vasconcelos, N. (2015). Bayesian model adaptation for crowd counts. In *ICCV*, pages 4175–4183.

Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., and Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. (2019). Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*.

Lizier, J. T. and Prokopenko, M. (2010). Differentiating information transfer and causal effect. *The European Physical Journal B*, 73(4):605–615.

LLC, T. D. S. C. (2016). Dark sky forecast api.

Lockerd, A. and Breazeal, C. (2004). Tutelage and socially guided robot learning. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 4, pages 3475–3480. IEEE.

Lowe, R., Noseworthy, M., Serban, I. V., Angelard-Gontier, N., Bengio, Y., and Pineau, J. (2017a). Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1116–1126.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. (2017b). Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390.

Lu, H. and Kawaguchi, K. (2017). Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*.

MacKay, D. J. (1992). The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Mazare, P.-E., Humeau, S., Raison, M., and Bordes, A. (2018). Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779.

McDuff, D. (2016). Discovering facial expressions for states of amused, persuaded, informed, sentimental and inspired. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 71–75. ACM.

McDuff, D. et al. (2016). Affdex sdk: A cross-platform real-time multi-face expression recognition toolkit. In *CHI*, pages 3723–3726. ACM.

Meeren, H., van Heijnsbergen, C., and de Gelder, B. (2005). Rapid perceptual integration of facial expression and emotional body language. *Proc. of the National Academy of Sci. of the USA*, 102(45):16518–16523.

Mehrabian, A. (2017). *Nonverbal communication*. Routledge.

Melis, A. P. and Semmann, D. (2010). How is human cooperation different? *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 365(1553):2663–2674.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miotto, R., Li, L., Kidd, B. A., and Dudley, J. T. (2016). Deep patient:

An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. *proceedings of ACL-08: HLT*, pages 236–244.

Mitchell, T. M. (1997). *Machine learning*. McGraw Hill.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Moerland, T. M., Broekens, J., and Jonker, C. M. (2018). Emotion in reinforcement learning agents and robots: a survey. *Machine Learning*, 107(2):443–480.

Mohamed, S. and Rezende, D. J. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 2125–2133.

Moturu, S. et al. (2011). Using social sensing to understand the links between sleep, mood, and sociability. In *Int. Conf. on Social Comput.*, pages 208–214. IEEE.

Mubaris, N. (2017). Support vector machines for classification.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062.

Murphy, K. (2012). *Mach. learning: a probabilistic perspective*. MIT press.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017). Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785.

Nautiyal, D. (2019). Underfitting and overfitting in machine learning.

Nosakhare, E. and Picard, R. (2019). Probabilistic latent variable modeling for assessing behavioral influences on well-being. In

Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2718–2726. ACM.

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.

Oudeyer, P.-Y. and Kaplan, F. (2006). Discovering communication. *Connection Science*, 18(2):189–206.

Oudeyer, P.-Y., Kaplan, F., et al. (2008). How can we define intrinsic motivation. In *Proc. of the 8th Conf. on Epigenetic Robotics*, volume 5, pages 29–31.

Oudeyer, P.-Y. and Smith, L. B. (2016). How evolution may work through curiosity-driven developmental process. *Topics in Cognitive Science*, 8(2):492–502.

Park, Y., Cho, J., and Kim, G. (2018). A hierarchical latent structure for variational conversation modeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1792–1801.

Partonen, T. (1996). Dopamine and circadian rhythms in seasonal affective disorder. *Medical hypotheses*, 47(3):191–192.

Patel, M., Asch, D., and Volpp, K. (2015). Wearable devices as facilitators, not drivers, of health behavior change. *Jama*, 313(5):459–460.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544.

Pearl, J. (2013). Structural counterfactuals: A brief introduction. *Cognitive science*, 37(6):977–985.

Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal inference in statistics: a primer*. John Wiley & Sons.

Peirce, R. et al. (2000). A longitudinal model of social contact, social support, depression, and alcohol use. *Health Psychology*, 19(1):28.

Pentland, A. (2004). Social dynamics: Signals and behavior. In *Int. Conf. on Developmental Learning*, volume 5.

Perolat, J., Leibo, J. Z., Zambaldi, V., Beattie, C., Tuyls, K., and Graepel, T. (2017). A multi-agent reinforcement learning model of common-pool resource appropriation. In *Advances in Neural Information Processing Systems*, pages 3643–3652.

Peters, J., Mülling, K., and Altun, Y. (2010). Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta.

Peysakhovich, A. and Lerer, A. (2017). Consequentialist conditional cooperation in social dilemmas with imperfect information. *arXiv preprint arXiv:1710.06975*.

Peysakhovich, A. and Lerer, A. (2018). Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2043–2044. International Foundation for Autonomous Agents and Multiagent Systems.

Picard, R. W. (2000). *Affective computing*. MIT press.

Poh, M. et al. (2010). A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *Biomedical Eng.*, 57(5):1243–1252.

Poh, M.-Z. (2011). *Continuous assessment of epileptic seizures with wrist-worn biosensors*. PhD thesis, MIT.

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

Provine, R. R. (2001). *Laughter: A scientific investigation*. Penguin.

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S., and Botvinick, M. (2018). Machine theory of mind. *arXiv preprint arXiv:1802.07740*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Ramirez, R. and Vamvakousis, Z. (2012). Detecting emotion from eeg signals using the emotive epoc device. In *International Conference on Brain Informatics*, pages 175–184. Springer.

Rashkin, H., Smith, E. M., Li, M., and Boureau, Y.-L. (2018). I know the feeling: Learning to converse with empathy. *arXiv preprint arXiv:1811.00207*.

Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, MA.

Ratey, J. (2008). *Spark: The revolutionary new science of exercise and the brain*. Hachette Digital, Inc.

Rawlik, K., Toussaint, M., and Vijayakumar, S. (2012). On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics: science and systems*.

Reinhardt, T. et al. (2012). Salivary cortisol, heart rate, electrodermal activity and subjective stress responses to the mannheim multicomponent stress test (mmst). *Psychiatry research*, 198(1):106–111.

Reis, H. and Gable, S. (2003). Toward a positive psychology of relationships.

Rendell, L., Boyd, R., Cownden, D., Enquist, M., Eriksson, K., Feldman, M. W., Fogarty, L., Ghirlanda, S., Lillicrap, T., and Laland, K. N. (2010). Why copy others? insights from the social learning strategies tournament. *Science*, 328(5975):208–213.

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.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Rokach, L. (2009). *Pattern classification using ensemble methods*. World Scientific.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., et al. (2019). Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433*.

Roose, K. (2019). YoutubeâĂŹs product chief on online radicalization and algorithmic rabbit holes. *The New York Times*.

Rosenstein, M. et al. (2005). To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, volume 898.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

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.

Salakhutdinov, R., Tenenbaum, J., and Torralba, A. (2013). Learning with hierarchical-deep models. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1958–1971.

Sanchez, B. (2018). Predicting iris flower species with k-means clustering in python.

Sano, A. (2015). *Measuring College Students' Sleep, Stress and Mental Health with Wearable Sensors and Mobile Phones*. PhD thesis, MIT.

Sano, A. et al. (2015a). Prediction of happy-sad mood from daily behaviors and previous sleep history. In *EMBC*. IEEE.

Sano, A. et al. (2015b). Recognizing academic performance, sleep quality, stress level, and mental health using personality traits, wearable sensors and mobile phones. In *Body Sensor Networks*.

Sano, A. and Picard, R. (2013a). Recognition of sleep dependent memory consolidation with multi-modal sensor data. In *Body Sensor Networks (BSN)*, pages 1–4. IEEE.

Sano, A. and Picard, R. (2013b). Stress recognition using wearable sensors and mobile phones. In *ACII*, pages 671–676. IEEE.

Sano, A. and Picard, R. (2014). Comparison of sleep-wake classification using electroencephalogram and wrist-worn multi-modal sensor data. In *EMBC*, pages 930–933. IEEE.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Schelling, T. C. (1973). Hockey helmets, concealed weapons, and daylight saving: A study of binary choices with externalities. *Journal of Conflict resolution*, 17(3):381–428.

Scherer, S., Morency, L.-P., Gratch, J., and Pestian, J. (2015). Reduced vowel space is a robust indicator of psychological distress: A cross-corpus analysis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4789–4793. IEEE.

Schmidhuber, J. (1991). Adaptive confidence and adaptive curiosity. In *Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2*. Citeseer.

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897.

Schultz, W. (2007). Behavioral dopamine signals. *Trends in neurosciences*, 30(5):203–210.

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. (2017). Generating focussed molecule libraries for drug discovery with recurrent neural networks. *arXiv preprint arXiv:1701.01329*.

Seligman, M. (2012). *Flourish: A visionary new understanding of happiness and well-being*. Simon and Schuster.

Senechal, T., McDuff, D., and Kaliouby, R. (2015). Facial action unit detection using active learning and an efficient non-linear kernel approximation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18.

Sequeira, P., Melo, F. S., Prada, R., and Paiva, A. (2011). Emerging social awareness: Exploring intrinsic motivation in multiagent learning. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–6. IEEE.

Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., et al. (2017a). A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., and Bengio, Y. (2017b). A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Shah, P., Hakkani-Tur, D., Liu, B., and Tur, G. (2018). Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Confer-*

ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pages 41–51.

Shcherbakov, O. and Batishcheva, V. (2014). Image inpainting based on stacked autoencoders. In *Journal of Physics: Conference Series*, volume 536, page 012020. IOP Publishing.

Shen, X., Su, H., Niu, S., and Demberg, V. (2018). Improving variational encoder-decoders in dialogue generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shi, W. and Yu, Z. (2018). Sentiment adaptive end-to-end dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1509–1519.

Shin, J., Xu, P., Madotto, A., and Fung, P. (2019). Happybot: Generating empathetic dialogue responses by improving user experience lookahead. *arXiv preprint arXiv:1906.08487*.

Shivhare, S. N. and Khethawat, S. (2012). Emotion detection from text. *arXiv preprint arXiv:1205.4944*.

Shum, M., Kleiman-Weiner, M., Littman, M. L., and Tenenbaum, J. B. (2019). Theory of minds: Understanding behavior in groups through inverse planning. *arXiv preprint arXiv:1901.06085*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

Singh, S. P., Barto, A. G., and Chentanez, N. (2004). Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1281–1288.

Srivastava, N. et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Stavropoulos, K. K. and Carver, L. J. (2013). Research review: social motivation and oxytocin in autism–implications for joint attention development and intervention. *Journal of Child Psychology and Psychiatry*, 54(6):603–618.

Stengel, R. F. (1986). *Stochastic optimal control*. John Wiley and Sons New York, New York.

Still, S. and Precup, D. (2012). An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131(3):139–148.

Storm, H. et al. (2000). The development of a software program for analyzing spontaneous and externally elicited skin conductance changes in infants and adults. *Clin. Neurophysiology*, 111(10):1889–1898.

Strouse, D., Kleiman-Weiner, M., Tenenbaum, J., Botvinick, M., and Schwab, D. (2018). Learning to share and hide intentions using information regularization. *arXiv preprint arXiv:1808.02093*.

Su, P.-H., Budzianowski, P., Ultes, S., Gasic, M., and Young, S. (2017). Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157.

Suhara, Y., Xu, Y., and Pentland, A. (2017). Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 715–724. International World Wide Web Conferences Steering Committee.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147.

Sutton and Barto (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tajima, S., Yanagawa, T., Fujii, N., and Toyoizumi, T. (2015). Untangling brain-wide dynamics in consciousness by cross-embedding. *PLoS computational biology*, 11(11):e1004537.

Taylor, S., Jaques, N., Chen, W., Fedor, S., Sano, A., and Picard, R. (2015). Automatic identification of artifacts in electrodermal activity data. In *EMBC*. IEEE.

Taylor, S. A., Jaques, N., Nosakhare, E., Sano, A., and Picard, R. (2017). Personalized multitask learning for predicting tomorrow's mood, stress, and health. *IEEE Transactions on Affective Computing*.

Thomas, P. and Brunskill, E. (2016). Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148.

Thomas, P., Theocharous, G., and Ghavamzadeh, M. (2015). High confidence policy improvement. In *International Conference on Machine Learning*, pages 2380–2388.

Thomaz, A. L. and Breazeal, C. (2007). Robot learning via socially guided exploration. *Development and Learning*, pages 82–87.

Thomaz, A. L. and Cakmak, M. (2013). Active social learning in humans and robots. *Social learning theory: Phylogenetic considerations across animal, plant, and microbial taxa, ed. KB Clark*, pages 113–28.

Tian, Y.-I., Kanade, T., and Cohn, J. F. (2001). Recognizing action units for facial expression analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 23(2):97–115.

Todorov, E. (2007). Linearly-solvable markov decision problems. In *Advances in neural information processing systems (NIPS)*, pages 1369–1376.

Tomasello, M. (2009a). *The cultural origins of human cognition*. Harvard university press.

Tomasello, M. (2009b). *Why we cooperate*. MIT press.

Truong, K. et al. (2014). Slide to x: unlocking the potential of smartphone unlocking. In *Human factors in comput. systems*, pages 3635–3644. ACM.

Tsuno, N. et al. (2005). Sleep and depression. *J. of Clin. Psychiatry*.

Umematsu, T., Sano, A., and Picard, R. (2019). Daytime data and lstm can forecast tomorrow's stress, health, and happiness. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*.

Valenza, G., Citi, L., Lanatá, A., Scilingo, E. P., and Barbieri, R. (2014). Revealing real-time emotional responses: a personalized assessment based on heartbeat dynamics. *Scientific reports*, 4:4998.

Valstar, M. et al. (2012). Meta-analysis of the first facial expression recognition challenge. *Systems, Man, and Cybernetics*, 42(4):966–979.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

van Schaik, C. P. and Burkart, J. M. (2011). Social learning and evolution: the cultural intelligence hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1567):1008–1016.

van Vugt, H., Hoorn, J., and Konijn, E. (2009). Interactive engagement with embodied agents: An empirically validated framework. *Comp. Animation and Virtual Worlds*, 20(2-3):195–204.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Veenhoven, R. (2008). Healthy happiness: Effects of happiness on physical health and the consequences for preventive health care. *Journal of happiness studies*, 9(3):449–469.

Venkatesh, A., Khatri, C., Ram, A., Guo, F., Gabriel, R., Nagar, A., Prasad, R., Cheng, M., Hedayatnia, B., Metallinou, A., et al. (2018). On evaluating and comparing conversational agents. *arXiv preprint arXiv:1801.03625*, 4:60–68.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408.

von Frisch, K. (1969). The dance language and orientation of bees. 5.

Vyzas, E. (1999). *Recognition of emotional and cognitive states using physiological data*. PhD thesis, MIT.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732.

Weininger, D. (1970). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. In *Proc. Edinburgh Math. SOC*, volume 17, pages 1–14.

Westefeld, J. and Furr, S. (1987). Suicide and depression among college students. *Professional Psychology: Research and Practice*, 18(2):119.

Wilhelm, F. H. and Roth, W. T. (1998). Taking the laboratory to the skies: Ambulatory assessment of self-report, autonomic, and respiratory responses in flying phobia. *Psychophysiology*, 35(5):596–606.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Wilson, A. et al. (2007). Multi-task reinforcement learning: a hierarchical bayesian approach. In *ICML*, pages 1015–1022. ACM.

Wilson, D. R. and Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural networks*, 16(10):1429–1451.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390.

Wong, J. and McGee, K. (2012). Frown more, talk more: effects of facial expressions in establishing conversational rapport with virtual agents. In *IVA*, pages 419–425.

Xia, R. and Liu, Y. (2015). A multi-task learning framework for emotion recognition using 2d continuous space. *IEEE Transactions on Affective Computing*.

Xie, J., Xu, L., and Chen, E. (2012). Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349.

Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., and Ma, W.-Y. (2017). Topic aware neural response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Xu, Y. et al. (1994). Wavelet transform domain filters: a spatially selective noise filtration technique. *Image Processing*, 3(6):747–758.

Xue, W., Huang, Z., Luo, X., and Mao, Q. (2015). Learning speech emotion features by joint disentangling-discrimination. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pages 374–379. IEEE.

Xue, Y. et al. (2007). Multi-task learning for classification with dirichlet process priors. *The Journal of Machine Learning Research*, 8:35–63.

Yang, Z., Metallinou, A., and Narayanan, S. (2014). Analysis and predictive modeling of body language behavior in dyadic interactions from multimodal interlocutor cues. *Multimedia*, 16(6):1766–1778.

Yu, C., Zhang, M., and Ren, F. (2013). Emotional multiagent reinforcement learning in social dilemmas. In *International Conference* *on Principles and Practice of Multi-Agent Systems*, pages 372–387. Springer.

Yu, N. (2007). Gradient methods for minimizing composite objective function. *CORE Discussion Papers*.

Zangróniz, R., Martínez-Rodrigo, A., Pastor, J., López, M., and Fernández-Caballero, A. (2017). Electrodermal activity sensor for classification of calm/distress condition. *Sensors*, 17(10):2324.

Zhang, B., Provost, E. M., and Essl, G. (2017). Cross-corpus acoustic emotion recognition with multi-task learning: Seeking common ground while preserving differences. *IEEE Transactions on Affective Computing*.

Zhang, C. and Zhang, Z. (2014). Improving multiview face detection with multi-task deep convolutional neural networks. In *Applications of Computer Vision*, pages 1036–1041. IEEE.

Zhao, Papangelis, and Cassell (2014). Towards a dyadic computational model of rapport management for human-virtual agent interaction. In *Intelligent Virtual Agents*, pages 514–527.

Zhao, T., Zhao, R., and Eskenazi, M. (2017). Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664.

Zhou, H., Huang, M., Zhang, T., Zhu, X., and Liu, B. (2018a). Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhou, L., Gao, J., Li, D., and Shum, H.-Y. (2018b). The design and implementation of xiaoice, an empathetic social chatbot. *arXiv preprint arXiv:1812.08989*.

Zhou, X. and Wang, W. Y. (2018). Mojitalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1128–1137.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.