

Copyright  
by  
Ruohan Zhang  
2021

The Dissertation Committee for Ruohan Zhang  
certifies that this is the approved version of the following dissertation:

**A Modular Attention Hypothesis for Modeling  
Visuomotor Behaviors**

Committee:

---

Dana Ballard, Supervisor

---

Peter Dayan

---

Mary Hayhoe

---

Alexander Huth

---

Peter Stone

**A Modular Attention Hypothesis for Modeling  
Visuomotor Behaviors**

by

**Ruohan Zhang**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2021

To my family.

## Acknowledgments

I owe the greatest thank to my academic advisors, Professor Dana Ballard and Professor Mary Hayhoe. I came to UT Computer Science Department with a bachelor's degree in psychology in 2012, and I struggled to find my place. I joined the Vision, Cognition, and Action VR Lab in Spring 2013 which turned out to be a beginning of a great journey. You have been always supportive of my research, career, and life plans.

I want to thank Professor Peter Stone, who taught me about robotics, reinforcement learning, and robotic soccer. Then I became a member of the UT Austin Villa RoboCup team in 2014 and had so much fun since then. Soccer was my first career goal when I was a little boy, and my hobby then was reading about robots in Asimov's books. I had no idea one day I could program robots to play soccer.

I am also very grateful for the other members of my dissertation committee whose feedback greatly improved the dissertation. I very much enjoyed Professor Peter Dayan's textbook on theoretical neuroscience. Professor Alexander Huth's Neural Computation class was one of the best classes I have ever had. As a computer scientist who is passionate about how the brain works, both of you showed me how to build a potential career at the intersection of two of my favorite scientific fields. You will always be an inspiration for my

future academic career.

I took many excellent courses while studying at UT. I would like to thank Professor Isil Dillig, Kristen Grauman, Risto Miikkulainen, Raymond Mooney, Yuke Zhu, and Mark Maxwell for teaching me everything.

I have been fortunate to have many amazing friends, colleagues, and collaborators. I would like to thank Yuchen Cui, Lin Guan, Suna Guo, Josiah Hanna, Xiangru Huang, Sariel Li, Bo Liu, Yuezhong Liu, Zhuode Liu, Patrick MacAlpine, Jacob Menashe, Karl Muller, Prabhat Nagarajan, Sanmit Narvekar, Akanksha Saran, Zhao Song, Matthew Tong, Faraz Torabi, Garrett Warnell, Calen Walshe, Jake Whritner, Ian Yen, Yue Yu, Luxin Zhang, Shiqi Zhang, and Yifeng Zhu. I very much enjoyed working and co-authoring with all of you. I especially want to thank Zhuode Liu, who started the Atari-HEAD project with me, which became a major part of this thesis.

Before beginning my Ph.D., I was very fortunate to have the mentorship of Professor Natalie Person, Katherine White, Chris Wetzel, Betsy Sanders, and Art Carden at Rhodes College. They gave me much guidance as I began learning how to do research.

I thank my parents, Lishan Huang and Lihua Zhang, as well as my other family members. I came to the United States 13 years ago. Your emotional support is very important to me. I would like to thank my friends, Yu-an Chen, Sariel Li, David Siu, and many others, for all your support and love these years.

Finally, to Edith Zeng and Twinkle the Dachshund, who have been

through everything with me in this journey, I would not be able to do all these without you.

Ruohan Zhang  
The University of Texas at Austin  
May 2021

# A Modular Attention Hypothesis for Modeling Visuomotor Behaviors

Publication No. \_\_\_\_\_

Ruohan Zhang, Ph.D.

The University of Texas at Austin, 2021

Supervisor: Dana Ballard

In this dissertation, we explore the hypothesis that complex intelligent behaviors, *in vivo*, can be decomposed into *modules*, which are organized in hierarchies and executed in parallel. This organization is similar to a multiprocessing architecture *in silico*. Biological attention can be viewed as a “process manager” that manages information processing and multiple computations. In this work, we seek to understand and model this modular attention mechanism for humans in a range of behavioral settings.

We explain this approach to understanding modular attention at three levels based on David Marr’s paradigm: the computation theory level, the representation and algorithm level, and the hardware implementation level. At the computation theory level, we propose that simple visuomotor behaviors can be broken down into modules that require attention for their execution. At the representation and algorithm level, we model human eye movements and

actions in a variety of visuomotor tasks. We collect and publish a large-scale, high-quality dataset of eye movements and actions of humans playing Atari video games. We study the active vision problem by jointly modeling human eye movements and actions, and compare how humans and artificial learning agents play these video games differently. We then propose a modular reinforcement learning model for modeling human subjects' navigation behaviors in a virtual-reality environment with multiple goals. We further develop a modular inverse reinforcement learning algorithm to efficiently estimate the subjective reward and discount factors associated with each behavioral goal.

At the implementation level, we propose a theoretical neuronal communication model named gamma spike multiplexing that allows the cortex to perform multiple computations simultaneously without crosstalk. The model explains how the modular attention hypothesis might be implemented in the biological brain.

The end goals of this work are to (1) build models to explain and predict observed human visuomotor behaviors and attention; (2) use these biologically inspired models to develop algorithms for better artificial learning systems.

# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>viii</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Figures</b>	<b>xvii</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
<b>Chapter 2. Background</b>	<b>10</b>
2.1 Markov Decision Process . . . . .	10
2.2 Reinforcement Learning . . . . .	11
2.3 Human-in-the-Loop Reinforcement Learning . . . . .	13
2.3.1 Imitation Learning . . . . .	13
2.3.1.1 Behavioral Cloning (BC) . . . . .	14
2.3.1.2 Inverse Reinforcement Learning (IRL) . . . . .	15
2.3.2 Learning from Human Guidance . . . . .	16
2.3.2.1 Learning from Human Evaluative Feedback . . . . .	18
2.3.2.2 Imitation from Observation . . . . .	20
<b>Chapter 3. Attentional Control</b>	<b>21</b>
3.1 Summary of Contributions . . . . .	23
3.2 Atari-HEAD Dataset . . . . .	25
3.2.1 Motivation . . . . .	25
3.2.2 Task Domains and Data Collection . . . . .	26
3.2.3 Semi-Frame-by-Frame Game Mode . . . . .	29
3.2.4 Dataset Statistics . . . . .	33
3.3 Gaze Modeling as Saliency Prediction . . . . .	34

3.3.1	Task Definition . . . . .	34
3.3.2	Baseline Model and Results . . . . .	37
3.3.3	Gaze Prediction with Additional Information . . . . .	40
3.3.4	Individual Gaze Differences . . . . .	44
3.4	Decision Modeling with Attention Information . . . . .	44
3.4.1	Task Definition . . . . .	44
3.4.2	Models and Results . . . . .	45
3.4.2.1	Behavioral Cloning . . . . .	45
3.4.2.2	Behavioral Cloning with Biologically Plausible State Representation . . . . .	46
3.4.2.3	Attention-Guided Imitation Learning (AGIL) . . . . .	47
3.4.3	Why Attention Helps . . . . .	51
3.5	Extending AGIL: Coverage-Based Gaze Loss . . . . .	53
3.5.1	Method . . . . .	55
3.5.1.1	Auxiliary Gaze Loss . . . . .	58
3.5.2	Other Techniques to Incorporate Gaze . . . . .	58
3.5.3	Experiments and Results . . . . .	59
3.5.3.1	CGL Improves Behavioral Cloning . . . . .	60
3.5.3.2	CGL Improves Behavioral Cloning from Observa- tion . . . . .	62
3.5.3.3	CGL Improves T-REX . . . . .	65
3.5.3.4	Best Performing Models for each Game . . . . .	65
3.5.3.5	Visualizing CGL Attention . . . . .	66
3.5.4	Reducing Causal Confusion with Human Attention . . . . .	68
3.5.5	Summary . . . . .	69
3.6	Human versus Machine Attention . . . . .	70
3.6.1	Background and Motivation . . . . .	70
3.6.2	Method . . . . .	72
3.6.2.1	Human Attention Data and Model . . . . .	73
3.6.2.2	Reinforcement Learning Agent and Attention Model . . . . .	74
3.6.2.3	Comparison Metrics . . . . .	75
3.6.3	Results . . . . .	76
3.6.3.1	The Effects of Learning . . . . .	77

3.6.3.2	The Effects of Discount Factors . . . . .	80
3.6.3.3	Failure States Analysis . . . . .	83
3.6.3.4	Unseen Data Analysis . . . . .	86
3.6.3.5	Other Atari Agents . . . . .	89
3.6.4	Discussion . . . . .	91
3.7	Human Attention-Guided Reinforcement Learning . . . . .	92
3.7.1	Attention-Guided Reinforcement Learning . . . . .	93
3.7.2	Guiding Reinforcement Learning with Evaluative Feedback and Attention Information . . . . .	97
3.8	Discussion, Related Work, and Future Work . . . . .	98
3.8.1	Related Work . . . . .	99
3.8.1.1	Related Work: Similar Datasets . . . . .	99
3.8.1.2	Related Work: Human Attention-Guided Imitation Learning . . . . .	100
3.8.1.3	Related Work: Human Attention in Robotics . . . . .	102
3.8.1.4	Related Work: Human versus Machine Attention . . . . .	104
3.8.2	Future Work . . . . .	105
3.9	Conclusion . . . . .	109
<b>Chapter 4. The Modularization Hypothesis</b>		<b>112</b>
4.1	Summary of Contributions . . . . .	116
4.2	Modular Reinforcement Learning and Inverse Reinforcement Learning . . . . .	117
4.2.1	Modular Reinforcement Learning . . . . .	117
4.2.2	Modular Inverse Reinforcement Learning . . . . .	122
4.2.2.1	Sparse Modular Inverse Reinforcement Learning . . . . .	124
4.3	Simulation Results . . . . .	126
4.3.1	Simulation: 2D Gridworld Navigation . . . . .	127
4.3.1.1	Modular vs. Standard Inverse Reinforcement Learning . . . . .	128
4.3.1.2	Sparse Modular Inverse Reinforcement Learning . . . . .	130
4.3.2	Simulation: Driving . . . . .	131
4.3.3	Action Selection in Modular Reinforcement Learning . . . . .	132
4.4	Human Navigation Experiment in Virtual Reality . . . . .	135

4.4.1	Experiment Design . . . . .	135
4.4.2	Results . . . . .	140
4.4.2.1	Qualitative Results and Visualization . . . . .	141
4.4.2.2	Between-task and Between-subject Differences . . . . .	145
4.4.2.3	Stability of Rewards and Discount Factors across Tasks . . . . .	147
4.4.2.4	Quantitative Results and Comparisons to Alter- native Models . . . . .	150
4.5	Discussion, Related Work, and Future Work . . . . .	153
4.5.1	Relation with other Reinforcement Learning Models . . . . .	154
4.5.2	Implications . . . . .	156
4.5.3	Limitations of the Model and Future Work . . . . .	158
4.6	Conclusion . . . . .	159
<b>Chapter 5. Neural Basis of Modularization and Attention</b>		<b>162</b>
5.1	Summary of Contributions . . . . .	164
5.2	Background and Motivation . . . . .	165
5.3	Gamma Spike Multiplexing Model . . . . .	168
5.3.1	Gamma Frequency Phase Coding Model . . . . .	168
5.3.2	Probabilistic Parallel Neuron Selection . . . . .	170
5.3.2.1	Sparse Coding Strategy . . . . .	170
5.3.2.2	Probabilistic Selection . . . . .	170
5.3.2.3	Parallel Selection . . . . .	171
5.3.3	Neural Multiplexing Model . . . . .	172
5.4	Simulation Results: A Sparse Coding Example . . . . .	175
5.4.1	Validating the Coding Algorithm . . . . .	178
5.4.2	Poisson Statistics and Parameter Sensitivity . . . . .	178
5.4.3	Observability of Gamma Oscillation . . . . .	181
5.5	Evidence from Neural Recording Data . . . . .	182
5.6	Discussion, Related Work, and Future Work . . . . .	184
5.6.1	Related Work . . . . .	187
5.6.2	Implications . . . . .	190
5.6.3	Future Work: Testing the Model . . . . .	193
5.7	Conclusion . . . . .	193

<b>Chapter 6. Conclusion</b>	<b>196</b>
6.1 The Modular Attention Hypothesis . . . . .	197
6.2 Human-in-the-Loop Reinforcement Learning . . . . .	199
6.3 Concluding Remarks . . . . .	205
<b>Appendices</b>	<b>207</b>
<b>Appendix A. Attentional Control</b>	<b>208</b>
A.1 Atari-HEAD Dataset . . . . .	208
A.2 Additional Results for Gaze Modeling . . . . .	210
A.3 Additional Results for Decision Modeling and Game Playing . . . . .	215
A.4 Additional Results for Coverage-Based Gaze Loss . . . . .	218
A.5 Additional Results for Human versus Machine Attention . . . . .	225
A.5.1 The Effects of Learning on Attention . . . . .	225
A.5.2 The Effects of Discount Factors on Attention . . . . .	226
A.5.3 Failure States Analysis . . . . .	226
A.5.4 Generalizing to Unseen Data . . . . .	227
<b>Appendix B. The Modularization Hypothesis</b>	<b>239</b>
B.1 Bayesian Inverse Reinforcement Learning . . . . .	239
<b>Appendix C. Neural Basis of Attention and Modularization</b>	<b>242</b>
C.1 Cost of Population Coding . . . . .	242
C.2 Sparse Coding and Neuron Selection . . . . .	243
C.3 A New Way of Interpreting Spike Data . . . . .	244
C.4 Neural Recording Data . . . . .	244
<b>Bibliography</b>	<b>250</b>
<b>Vita</b>	<b>310</b>

## List of Tables

3.1	A comparison of human scores for 20 Atari games across datasets	32
3.2	Quantitative results of predicting human gaze across eight games	43
3.3	Average (across 20 games) improvement over the T-REX baseline	65
3.4	A summary of the best game scores obtained using different imitation learning algorithms . . . . .	67
4.1	Estimated rewards and discount factors comparing to the ground truth for the six modules in the 2D gridworld experiment . . .	128
4.2	Estimated rewards and discount factors for the car, road, and target modules, for the nice driver and the aggressive driver .	132
4.3	One-way ANOVA for individual differences in reward between subjects and across task instructions . . . . .	147
4.4	Task-relevant module rewards and discount factors are transferable across task conditions . . . . .	149
4.5	Evaluation of the modular agent’s performance compared with baseline agents, measured by the average angular difference (in degrees) compared to actual human decisions . . . . .	152
5.1	Parameter values used in the simulations . . . . .	178
A.1	Quantitative results of predicting human gaze across 20 games	212
A.2	Behavior matching accuracy of different models . . . . .	216
A.3	Game scores of game agents trained using different sources of data	217
A.4	Game scores obtained when using 15-minute human demonstration data to train the behavioral cloning agents . . . . .	218
A.5	Game scores obtained when using all 300-minute human demonstration data to train the behavioral cloning agents . . . . .	219
A.6	Game scores obtained when using 15-minute human demonstration data to train the BCO agents . . . . .	220
A.7	Game scores obtained when using 300-minute human demonstration data to train the BCO agents . . . . .	221

A.8	Game scores obtained when using 30-minute human demonstration data to train the T-REX agents . . . . .	222
A.9	Game scores obtained when using 300-minute human demonstration data to train the T-REX agents . . . . .	223
A.10	Causal confusion study results . . . . .	224

## List of Figures

1.1	The modular attention hypothesis . . . . .	4
3.1	Project schematic for the Atari-HEAD dataset . . . . .	27
3.2	State-action mismatch in game Breakout . . . . .	30
3.3	Inattentive blindness in game Freeway . . . . .	30
3.4	Scanpath and reaction time in game Frostbite . . . . .	31
3.5	The gaze prediction network . . . . .	38
3.6	Gaze prediction results measured using four standard metrics, averaged across 20 games . . . . .	38
3.7	Example gaze prediction results for four Atari games . . . . .	39
3.8	The gaze model predicts gaze behaviors that are difficult to capture using hand-defined features . . . . .	40
3.9	The three-channel gaze prediction network . . . . .	41
3.10	A behavioral cloning network for predicting human actions . . . . .	46
3.11	The original game frames for Atari Seaquest with red circles indicating the gaze position . . . . .	47
3.12	Action prediction accuracy for models trained using foveated images with different visual angles . . . . .	48
3.13	The AGIL policy network architecture for predicting human actions . . . . .	49
3.14	Incorporating gaze model learned from humans improves imitation learning algorithm’s performance in terms of behavior matching accuracy . . . . .	50
3.15	Incorporating gaze model learned from humans improves imitation learning algorithm’s performance in terms of game scores . . . . .	52
3.16	Human gaze information helps the learning agent correctly infer the underlying reason for the chosen action . . . . .	53
3.17	Our auxiliary gaze loss (CGL) guides a convolutional network to focus on parts of the state space which the human attends to . . . . .	57
3.18	Average (across 20 games) percentage improvement over the BC baseline . . . . .	61

3.19	Average (across 20 games) improvement over the BCO baseline	63
3.20	Motion in the visual game state . . . . .	64
3.21	CGL guides a convolutional network to focus on parts of the state space which the human attends to . . . . .	68
3.22	Confounded states with past actions to test reduction of causal confusion with CGL . . . . .	69
3.23	Changes in human and RL attention similarity across learning time steps . . . . .	78
3.24	Attention of RL agents changes during learning and becomes more human-like . . . . .	79
3.25	Changes in human and RL attention similarity across different discount factors . . . . .	81
3.26	Effect of different discount factors on Ms.Pac-Man and Seaquest agents' attention . . . . .	82
3.27	RL vs. human attention in states where RL agents made mistakes	84
3.28	How attention similarities change in failure states compared to normal states . . . . .	85
3.29	Human versus RL attention in states that RL agents have not seen	88
3.30	The relation between the similarity with human attention and algorithm's performance . . . . .	90
3.31	The effect of attention on learning to play a simple game called "Catch". Attention helps the most when significant amount of noise is presented and perception itself is a difficult problem. . . . .	96
4.1	The concept of modular reinforcement learning illustrated using value surfaces . . . . .	121
4.2	Maximum likelihood modular inverse reinforcement learning . . . . .	122
4.3	Part of the 2D gridworld test domain . . . . .	127
4.4	Modular IRL vs Bayesian IRL vs sparse modular IRL on sample efficiency . . . . .	129
4.5	The simulated driving environment . . . . .	131
4.6	Test domain for action selection in modular RL . . . . .	134
4.7	The action selection strategies we proposed have similar performance in the gridworld . . . . .	136
4.8	The virtual-reality human navigation experiment with motion tracking . . . . .	137

4.8	Bird's-eye view of human trajectories and agent trajectory clouds across different subjects . . . . .	144
4.9	Average normalized rewards for each subject under different task instructions . . . . .	146
4.10	Normalized average estimated rewards and discount factors across different task instructions . . . . .	148
4.11	Average number of targets collected/obstacles hit when different models perform the navigation task across all trial . . . . .	151
4.12	Modular reinforcement learning vs. hierarchical reinforcement learning . . . . .	156
5.1	Gamma frequency phase coding model . . . . .	169
5.2	Parallel coding model . . . . .	172
5.3	Neural multiplexing model . . . . .	174
5.4	A sparse coding example . . . . .	176
5.5	The GSM model generates spikes that appear Poisson distributed in simulation . . . . .	179
5.6	Gamma oscillation is less visible when performing trial averaging	180
5.7	Somatic membrane potential oscillations are modulated by visual stimulus . . . . .	185
6.1	The state space representation provided to different reinforcement learning algorithms for Atari Seaquest game. . . . .	200
6.1	Human-agent-environment interaction diagrams of five learning frameworks . . . . .	204
A.1	20 Atari 2600 games were used to collect human gaze and action data . . . . .	209
A.2	Gaze prediction learning curve for eight games . . . . .	213
A.3	Correlation coefficient matrices of the gaze network when trained on one subject and tested on another subject . . . . .	214
A.4	Human and RL saliency maps become more similar over training time steps in game Breakout . . . . .	228
A.5	Human and RL saliency maps become more similar over training time steps in game Freeway . . . . .	228
A.6	Human and RL saliency maps become more similar over training time steps in game Frostbite . . . . .	229

A.7	Human and RL saliency maps become less similar at first, and then become more similar over training time steps in game Ms.Pac-Man . . . . .	230
A.8	Human and RL saliency maps become more similar over training time steps in game Montezuma’s Revenge . . . . .	230
A.9	Human and RL saliency maps becomes more similar according to the KL metric in game Seaquest . . . . .	231
A.10	The RL agent’s attention is most similar to human’s when $\gamma = 0.9$ in game Breakout . . . . .	231
A.11	The RL agent’s attention is most similar to human’s when $\gamma = 0.9, 0.99$ in game Freeway . . . . .	232
A.12	The RL agent’s attention is most similar to human’s when $\gamma = 0.7$ (CC) or $0.5$ (KL) in game Frostbite . . . . .	232
A.13	The RL agent’s attention is most similar to human’s when $\gamma = 0.9$ in game Ms.Pac-Man . . . . .	233
A.14	The RL agent’s attention is most similar to human’s when $\gamma = 0.99$ in game Montezuma’s Revenge . . . . .	233
A.15	The RL agent’s attention is most similar to human’s when $\gamma = 0.5$ in game Seaquest . . . . .	234
A.16	Games in which human attention and RL agents’ attention are more different in the failure states than the normal states . . .	235
A.17	Games in which human attention and RL agents’ attention are more similar in the failure states than the normal states . . . .	236
A.18	Games in which human attention and RL agents’ attention are more different in unseen states than the normal states . . . . .	237
A.19	Games in which human attention and RL agents’ attention are more similar in the unseen states than the normal states . . .	238
B.1	Example of data discretization in the VR navigation experiment.	240
C.1	Estimating the cost of population code that assumes rate code	243
C.2	The effect of the number of coding neurons on the accuracy of the parallel probabilistic coding algorithm . . . . .	245
C.3	Given an image patch, coding neurons are selected probabilistically	246
C.4	GSM provides a new way of interpreting cell recording data .	247
C.5	An example patch-clamp data from a single trial . . . . .	249

# Chapter 1

## Introduction

Computational modeling of human vision dates from Marr's seminal proposal that divided the theories into *problem specification*, *algorithm*, and *biological implementation* [199]. For more than three decades, Marr's three levels of analysis have become one of the standard frameworks for understanding information-processing systems. This organization decouples the functional model of experimental observations from a complete account of all its extremely complex implementation details. Such abstraction allows Marr's framework to be applied to different research fields and facilitates the communication between these fields. For example, even though biological brains and electronic computers are vastly different at the hardware implementation level, at more abstract levels the underlying computational problems, representations, and algorithms are indeed similar, or at least comparable in many cases. At these abstract levels, knowledge gained in one field could greatly inspire related research fields.

In this work, we seek to answer the following research question using this approach:

How does the brain learn and make decisions to achieve behavioral

goals in an information-rich environment, with limited cognitive resources?

Humans accomplish a variety of complex daily sensory-motor tasks with ease and grace. Many scenarios require one to perform multiple tasks simultaneously or in a short time window. How does the biological brain manage different processes to solve different tasks in this setting? At the most abstract level, we try to answer the question of how these tasks are defined, and how they are related to each other. Level III of Marr's paradigm concerns the following question:

What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out [199]?

Many behaviors of humans and animals are reward-seeking, i.e., the goal of the computation process is to accomplish certain tasks to maximize reward. However, multiple potential sources of reward are often presented simultaneously. For example, when crossing a road, a person must simultaneously determine the direction of heading, avoid tripping over the curb, and locate other pedestrians or vehicles. Each of the goals could be associated with a unique reward. In modeling such behaviors, one can understand the overall goal of computation through understanding its parts – the individual reward associated with each process, and its relative importance compared to rewards associated with other processes.

Based on this observation, we propose the following hypothesis about how the brain solves complex visuomotor tasks. We name this cognitive model the *modular attention* hypothesis, shown in Fig. 1.1. This hypothesis suggests that complex behaviors can be broken down into multiple *modules*, each of which requires specific sensory information, has different goals (in terms of rewards), and prefers different actions [24]. The different modules are denoted with different colors and numbers. The different lengths indicate that modules can exhibit different numbers of states and finish at different times. In a given context (episode), only a few modules are activated, and irrelevant ones are kept inactive. While a small number of modules can be simultaneously active, the composition of the modules can be changed quickly, allowing for a rich tapestry of different modules to be active over time. The modal fixation interval of 200 to 300 milliseconds provides a logical time constant for module switching.

What is the algorithm that manages information flow that feeds necessary information to each module, and decides their priorities? The biological *attentional control* mechanism can be viewed as a "process manager" or a scheduler that manages information processing and multiple computations. It has two main roles in this model. Firstly, it is a scheduler that determines which modules should be activated given the current high-level behavioral goal and environment context. In the human cognition literature, this can be thought of as *executive control* – transitions from one behavior to the next will depend on varying degrees on learned behavioral sequences and complex decision processes that take into account the momentary behavioral goals [193, 260].

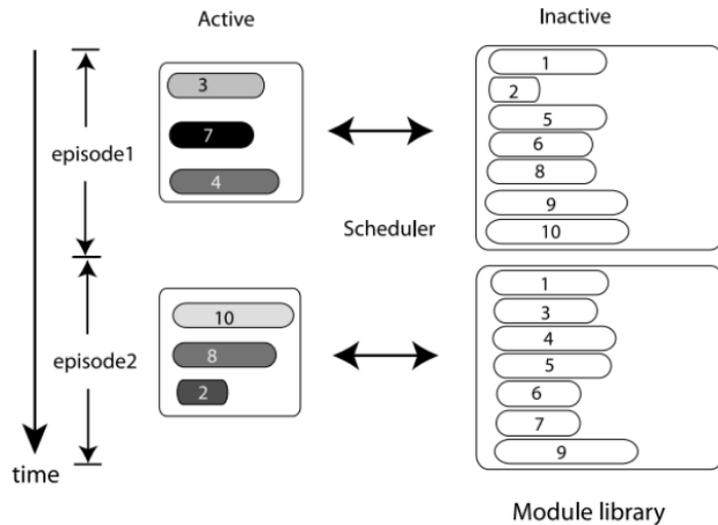


Figure 1.1: The modular attention hypothesis. Figure is adapted from [23]. Complex behaviors can be broken down into multiple *modules*, each of which requires specific sensory information, has different goals (in terms of rewards), and prefers different actions. The different modules are denoted with different colors and numbers. The different lengths indicate that modules can exhibit different numbers of states and finish at different times. In a given context (episode), only a few modules are activated, and irrelevant ones are kept inactive.

Secondly, it is a resource manager that controls the sensors and actuators of the body. Embodied agents (humans or AIs) have physical constraints in their sensory-motor systems therefore active modules may compete for the limited resources while trying to accomplish their own goals. Attention allocates the resources based on the urgency of these modules.

The modular attention model is abstract. At Level II of Marr’s paradigm, we will flesh out the model through modeling human visuomotor behaviors:

In particular, what is the representation for the input and output,

and what is the algorithm for the transformation [199]?

A theoretical basis for modeling reward-seeking behaviors is reinforcement learning (RL). An important factor that makes standard RL difficult in modeling natural behaviors is its sophistication and resulting computational burden as a model for general reward-seeking behaviors. Therefore the standard RL must be extended to make computation tractable. An extension of standard RL named *modular reinforcement learning* utilizes divide-and-conquer as an approximation strategy [258, 265, 288]. The modular RL takes the statistical structure present in the environment, decomposes a task into *modules* where each RL module solves a subgoal of the original task.

Several studies have explored the plausibility of a modular architecture for natural visually guided behavior where complex tasks can be broken down into concurrent execution of these modules [23, 96, 162, 307]. In the example of walking across the street, each particular behavioral subgoal such as avoiding obstacles can be treated as an independent module. This leads to a view of the human brain as the centralized arbitrator that divides and coordinates these modules in a hierarchical manner. The current investigation explores the modular architecture in more detail.

At the representation and algorithm level, to achieve a particular goal, each module requires some particular visual input about the state of the world and may need to compromise with each other to make an appropriate decision at the moment. For example, video games often involve multiple

tasks at any given moment. Solving each task requires rapidly moving the eyes to perceive relevant information in a dynamic environment, memorizing important objects' locations, predicting the future state of the moving targets, and making decisions that maximize the expected reward. All of these could be cognitively demanding and one must allocate resources on the most urgent task and associated computation process. The visual attention mechanism achieves this goal by allowing players to identify, process, and respond to a reduced set of task features that carries important information.

Luckily, in visuomotor tasks, the attention is partially revealed by human eye movements (*gaze*). It is well known that human eye movements are closely related to task rewards [116]. Evidence has shown that human gaze can be considered as an overt behavioral signal that encodes a wealth of information about both the motivation behind an action and the anticipated reward of an action [116, 117, 155]. Measuring and modeling attention via eye movements could reveal a rich amount of information about underlying computation processes and decision making. Therefore, at the second level of analysis, we hope to model human visual attention given the eye movement data in complex tasks.

At the implementation level of Marr's paradigm, one crucial question for understanding the cortex's neural networks is: Can cortical circuits be working independently at the same time hence modules can be implemented? There are several reasons to suspect that this must be possible. One is known as the binding problem and is remembered as the "red square blue circle problem." If,

in the cortex, "red" neurons and "blue" neurons are active along with "square" neurons and "circle" neurons, how does the cortex distinguish the one color-shape pairing from the other? This problem implies that the cortical network must be able to perform modularization in some way, and we try to describe a neural processing model that is capable of doing this. A complete answer to this question involves a more comprehensive brain model, here we address the implementation problem from a neuronal communication perspective.

Recent advances in the field of artificial intelligence (AI) have provided adequate tools for our mission. Testing the modular attention hypothesis at levels II and I involves analyzing a large amount of behavioral or neural data of humans and animals, which requires powerful data processing tools and computational models. Biologically inspired AI designs, through reverse-engineering the human brain, have given birth to several major breakthroughs in the field of AI, including artificial neural networks and reinforcement learning. They belong to a research field of machine learning, in which statistical learning algorithms allow machines to learn from data. These tools naturally suit our purpose because many of them were initially developed to explain intelligent behaviors *in vivo*, yet they can be deployed *in silico* to take the advantage of the computational speed of modern computers.

Although modern artificial intelligence (AI) systems have achieved phenomenal performance in specific visuomotor tasks like board games [280] and video games [209], they are often designed for a particular task. At level I, biological brain and machine hardware are inherently different. However, at

level II, the underlying mathematical principles, i.e., the *software* that defines general intelligence, may be similar. This opens the possibility that the AI may benefit from studying human and animal cognition, as shown by the recent progress in biologically inspired AI designs. The proposed work follows this trend and aims to discover the mechanism of intelligence by studying biological brains and behaviors and applying these insights to AI. Through building AI systems, we also gain a deeper understanding of the computational models while attempting to reproduce human-level intelligence. This leads to another focus of this work, that is:

How can we improve current artificial intelligence (AI) by studying these mechanisms of the brain, so that AIs can cope with the complexity in the real world?

This work is organized as follows: We will start with level II and then move to level I. At the representation and algorithm level, we study the active vision problem by jointly modeling human gaze and actions in several visuomotor tasks, such as playing video games [359, 363, 268, 352]. Also at level II, we propose a modular reinforcement learning model for understanding human subjects' behaviors in an environment with multiple goals and rewards. We further develop a modular inverse reinforcement learning algorithm to estimate subjective reward associated with each goal [367]. At the implementation level, we propose a theoretical neuronal communication model named gamma spike multiplexing that attempts to explain how the cortex performs multiple

computations simultaneously without crosstalk, and how the modular attention hypothesis may be implemented by a biological brain [20, 355].

# Chapter 2

## Background

### 2.1 Markov Decision Process

We first introduce mathematical models and notations that will be used throughout this work. A standard reinforcement learning model is formalized as a Markov decision process (MDP). The MDP models the interaction between the environment and a decision-maker which will be referred to as an agent. Formally, an MDP is defined as a tuple  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$  [298, 299], where:

- $\mathcal{S}$  is a finite set of environment states. Let  $s_t$  denote the agent's state at discrete time step  $t$ . The state encodes relevant information for an agent's decision.
- $\mathcal{A}$  is a finite set of available actions. Let  $a_t$  be the action agent chooses to take at time  $t$ . The agent interacts with the environment by taking an action in its observed state.
- $\mathcal{P}$  is the state transition function which specifies the probability  $P(s'|s, a)$ , i.e., the probability of entering state  $s'$  when agent takes action  $a$  in state  $s$ . The state transition function describes the dynamics of the environment that are influenced by an agent's action.

- $\mathcal{R}$  is a reward function.  $r_t$  denotes the scalar reward agent received at time step  $t$ .
- $\gamma \in [0, 1]$  is a discount factor. The agent values future rewards less than an immediate reward, therefore future rewards are discounted by parameter  $\gamma$  at every discrete time step.  $\gamma = 0$  indicates that the agent is myopic and only seeks to maximize the immediate reward.
- $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$  is called a policy of the agent, which specifies the probability of chosen each action in each state.

## 2.2 Reinforcement Learning

In machine learning, the purpose of a reinforcement learning algorithm is to find an optimal policy  $\pi^*$  that maximizes the expected long-term cumulative reward. One could optimize  $\pi$  directly, while alternatively many of the algorithms are based on value function estimation, i.e., estimating the state value function  $V^\pi(s)$  or the action-value function  $Q^\pi(s, a)$ .

The state value function defined for a given policy  $\pi$  is given by [299]

$$V^\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s \right] \quad (2.1)$$

A corresponding action value function,  $Q^\pi(s, a)$ , also exists and is given by

$$Q^\pi(s, a) = E_\pi [R(s_t, a_t) + V^\pi(s_{t+1}) \mid s_t = s, a_t = a] \quad (2.2)$$

and the advantage function  $A^\pi(s, a)$ , is defined as

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s). \quad (2.3)$$

The state value function  $V^\pi(s)$  measures the expected cumulative reward to be in a particular state  $s$  and following policy  $\pi$  afterward. The action-value function  $Q^\pi(s, a)$  defines the same quantity but for taking a particular action  $a$  when in state  $s$  and following policy  $\pi$  afterward. Given the Q-value function, it is convenient for an agent to select the action that maximizes expected future returns. The advantage function tells us the relative gain (“advantage”) that could be obtained by taking a certain action compared to the average action taken at that state [323].

Many successful RL algorithms that seek to estimate these quantities directly have been developed, including Q Learning [325] and advantage actor-critic (see, e.g., [299]). For example, Q Learning techniques seek to learn the state-action value function for the optimal policy,  $Q^{\pi^*}(s, a)$ , and the policy is then given by  $\pi^*(s) = \arg \max_a Q^{\pi^*}(s, a)$ . Nowadays, deep neural networks are often used as function approximators to estimate and optimize  $\pi$ ,  $V$ , and  $Q$ .

An important challenge in RL is to balance exploration vs. exploitation when an agent selects its action. Exploration allows the agent to improve its current knowledge. Exploitation chooses the greedy action to maximize reward by exploiting the agent’s current knowledge. A simple strategy ( $\epsilon$ -greedy) chooses a random action with probability  $\epsilon$  and chooses the greedy action (the action with the highest Q value) with probability  $1 - \epsilon$  [299]. A more sophisticated strategy uses a Boltzmann distribution for selecting actions based

on the current estimate of Q function [299]:

$$P(a|s, Q, \tau) = \frac{e^{Q(s,a)}/\tau}{\sum_{a' \in \mathcal{A}} e^{Q(s,a')}/\tau} \quad (2.4)$$

where  $\tau$  is a temperature constant that controls the exploration rate.

## 2.3 Human-in-the-Loop Reinforcement Learning

### 2.3.1 Imitation Learning

The standard imitation learning setting can be formulated as  $\text{MDP} \setminus \mathcal{R}$ , i.e. there is no reward function  $\mathcal{R}$  available. Instead, a learning agent (the *imitator*) records expert (the *demonstrator*, could be expert humans or artificial agents) demonstrations in the format of state-action pairs  $\{(s_t, a_t^*)\}$  at each timestep, and then attempts to learn the expert policy. The learning objective is: Given a state  $s_t$ , learn to predict expert action  $a_t$ , i.e., learn policy  $\pi(s, a)$ .

One approach is for the agent to learn to mimic the demonstrated policy using supervised learning, which is known as behavioral cloning [16]. A second approach to imitation learning is called inverse reinforcement learning (IRL) [2] which involves learning a reward function based on the demonstration data and learning the imitation policy using RL with the learned reward function. These two approaches constitute the major learning frameworks used in imitation learning. Comprehensive reviews of these two approaches can be found in [9, 140, 227, 12, 85].

### 2.3.1.1 Behavioral Cloning (BC)

Behavioral cloning [238, 16] is one of the main methods to approach an imitation learning problem. The agent receives as training data both the encountered states and actions of the demonstrator, then uses supervised learning techniques such as classification or regression to estimate the demonstrator’s policy. This method is powerful in the sense that it is capable of imitating the demonstrator immediately without having to interact with the environment, and it has been successfully applied in many application domains. For instance, it has been used to train a quadrotor to fly down a forest trail [101]. There, the training data consists of images of the forest trail gathered by a camera mounted to the quadrotor and labeled with the actions that the demonstrating quadrotor used. The policy is modeled as a convolutional neural network classifier, and trained using supervised learning. In the end, the quadrotor managed to fly down the trail successfully. BC has also been used in autonomous driving [37]. The training data is acquired using a human demonstrator, and a convolutional neural network is trained to map raw pixels from a single front-facing camera directly to platform steering commands. After training, the vehicle was capable of driving in traffic on local roads. BC has also been successfully used to teach robotic manipulators complex, multi-step, real-world tasks using kinesthetic demonstrations [221].

One of BC’s major drawbacks is potential performance degradation due to the well-studied compounding error caused by covariate shift [255, 256], i.e., that training and testing data distribution mismatch results in deviation of the

learned behavior from the demonstration [308]. [256] proposed an interactive training method to correct the shift called DAgger (Dataset Aggregation) which attempts to bring the distribution of demonstration data closer to that of the learned behavior. It does so by collecting demonstration data on the states observed by the imitator at each iteration. Retraining the policy on the aggregated dataset ultimately prevents the imitator from deviating from the demonstration behavior.

### **2.3.1.2 Inverse Reinforcement Learning (IRL)**

Inverse reinforcement learning [2, 373] is a second category of imitation learning. IRL techniques seek to learn a reward function that has the maximum value for the demonstrated actions. The learned reward function is then used in combination with RL methods to find an imitation policy. To be more specific, most IRL algorithms first initialize a random policy. Next, the agent executes that policy in the environment to collect state-action data, and then the algorithms estimate the expert’s reward function based on the data generated by the policy and the demonstration data. Finally, standard RL algorithms are used to learn an optimal policy for that reward function. The process of reward learning and policy learning is repeated until the agent policy becomes sufficiently close to the demonstrator’s policy. Like BC techniques, IRL methods usually assume that state-action pairs are available [89], and also that the reward is a function of both states and actions. The algorithms developed in this category have shown impressive results in a variety of tasks

such as autonomous helicopter aerobatics [1], robot object manipulation [89], and autonomous navigation in complex unstructured terrains [279], etc.

One major drawback of most algorithms developed for IRL is that at each iteration, they have to solve a complete RL problem to find an optimal policy given the currently estimated reward function which is computationally very expensive. However, the learned policies are often more robust than the policies learned by BC algorithms as they do not suffer from the covariate shift problem. This shift does not happen in the case of IRL because the agent can interact with the environment while training and the distribution mismatch diminishes during the process.

The primary focus of RL has been on forward models that, given reward signals, can learn to produce policies, which specify action choices when immersed in an environment state. IRL has the potential to be useful in modeling human behaviors, since by estimating the reward one can provide an answer for Marr’s question at level III. Using IRL, a behavioral model can be quantitatively evaluated by comparing human behaviors with reproduced behaviors by an artificial agent trained using the RL model with the estimated reward function.

### **2.3.2 Learning from Human Guidance**

Both paradigms described above (i.e., RL and IL) have been used with remarkable success [209, 280, 183, 282, 281, 147, 320], especially when combined with deep learning [177] to solve challenging sequential decision-making tasks.

While reward functions and explicit action demonstrations currently represent the most common ways in which humans specify tasks for artificial learning agents to perform, recent years have seen a great deal of research energy devoted to studying alternative ways in which humans might perform task specification.

In general, these alternatives are focused on more diverse and creative ways of providing input than the two methods described above, and so we explicitly refer to the resulting types of input as *human guidance*. Because human guidance is less direct compared to specified reward functions or explicit action demonstrations, effort to leverage it has led to several new research challenges in the machine learning community.

There are many reasons for the recent interest in utilizing human guidance. One reason is the relative ease with which several forms of human guidance can be collected. For some tasks, it may be exceedingly difficult for a human trainer to specify a reward function or provide an action demonstration since both require some level of training and skill that the human may not possess. However, it may still be possible for the human to guide the learning agent. As an analogy from human learning, consider the sports coach that provides guidance in the form of feedback on professional athlete performance. Even though the coach typically can not explicitly demonstrate the skill to be performed at the same skill or performance level as the athlete, their feedback is often useful to the athlete. In these cases, the availability of guidance may even help the learner achieve greater final task performance than if an action demonstration alone was provided.

Another reason for the research community’s interest in studying machine learning from human guidance lies in the utility of human guidance as a supplemental training signal that can increase the speed of task learning. That is, even in cases for which a reward signal or an action demonstration is available, if the learning agent can leverage available human guidance, the overall amount of time it takes to arrive at an acceptable behavior policy can be greatly reduced compared to if the guidance had not been used at all.

### **2.3.2.1 Learning from Human Evaluative Feedback**

When training RL agents, instead of providing action demonstrations, it is also common to use other forms of human guidance signals. Here we discuss one of the most natural forms of human guidance that have been studied: *evaluative feedback*. Proposed paradigms for learning from evaluative feedback typically involve human trainers watching artificial agents attempt to execute tasks and those humans providing a scalar signal that communicates the desirability of the observed agent behavior. Using this type of human guidance, the learning problem for the agent is that of determining how to adjust its policy such that its future behavior becomes more desirable to the human.

Evaluative feedback is an attractive form of human guidance due to the relative ease with which humans can provide it. For example, for cases in which the human trainer cannot provide a demonstration of the task (because, e.g., the task is too difficult), the human typically still knows what constitutes

good behavior and can therefore provide evaluative feedback. Moreover, even when the human can provide a demonstration, providing additional evaluative feedback during the learning process may allow the artificial agent to achieve a task performance that exceeds that of the human demonstrator.

Several methods interpret the feedback signal as a value-like quantity [164, 196]. Intuitively, this interpretation amounts to assuming that the human feedback provides a rating of the agent’s current decision with respect to some forecast of future behavior. One such technique is the TAMER algorithm (training an agent manually via evaluative reinforcement) [164], in which it is assumed that the human has in mind a desired policy  $\pi_H$ , and the feedback given at a time instant  $t$ ,  $H(s_t, a_t)$  roughly corresponds to  $Q^{\pi_H}(s_t, a_t)$  (defined in Eq. 2.2). TAMER agents use supervised learning with all the feedback collected up to time  $t$  to calculate the current estimate of  $H$ ,  $\hat{H}$ , e.g., through minimizing a standard squared loss [324]:

$$\hat{H}^* = \arg \min_{\hat{H}} \sum_t \left[ \hat{H}(s_t, a_t) - H(s_t, a_t) \right]^2 \quad (2.5)$$

Then the agent acts, in the next state, according to the policy

$$a_{t+1} = \arg \max_a \hat{H}^*(s_{t+1}, a) \quad (2.6)$$

in a fashion similar to Q Learning since we interpret  $\hat{H}^*$  as an approximation for  $Q^{\pi_H}$ .

Notably, several have studied combining human-provided evaluative feedback with existing reward functions [58, 165, 166, 8] to augment reinforcement learning.

### 2.3.2.2 Imitation from Observation

Imitation from observation (IfO) [309] is the problem of learning directly by observing a trainer performing the task. The learning agent only has access to state demonstrations (e.g. visual observations) of the trainer. Using this type of human guidance, the learning problem for the agent is to learn a policy from the state sequences demonstrated by the human.

This framework is different from conventional imitation learning in the sense that it eschews the requirement for action labels in demonstrations. Removing this constraint enables imitating agents to use a large amount of previously ignored available demonstration data such as videos on YouTube. The ultimate goal in this framework is to enable agents to utilize the existing, rich amount of demonstration data that do not have action labels, such as the human guidance provided through online videos of humans performing various tasks.

## Chapter 3

### Attentional Control

The sensory input constitutes a high-dimensional state space for decision making, given an environment with rich information that animals and humans interact with. An important assumption we made about task state space is *sparsity*: Only a subset of state features is needed for accomplishing a particular subtask, or a module (see Chapter 4). Hence the transformation algorithm is not a function that directly maps a complete, high-dimensional state space to decisions. Rather, at the perception level, the transformation algorithm actively selects a subset of visual features that are relevant to the current task. The active nature of vision has long been recognized in the human perception literature, but the computational tools to explore have been lacking. Development of these tools can be traced to the computational literature by Bajcsy [17]:

"Most past and present work in machine perception has involved extensive static analysis of passively sampled data. However, it should be axiomatic that perception is not passive, but active. Perceptual activity is exploratory, probing, searching; percepts do not simply fall onto sensors as rain falls onto ground."

This quote has at least two new ideas. One is that perceptions are not passive, but are ordered up by an active process of the perceiver. Another is that rather than a complication, active perception provides additional constraints for the perceiver. This outstanding intelligent mechanism of humans is often referred to as *selective attention* – the ability to allocate cognitive resources to important things. Posner [239] distinguishes overt vs. covert attention. The former is directly revealed by eye movements which consciously focus the eye on the visual stimulus, while the latter is achieved by mentally shifting one’s focus without moving one’s eyes [239].

Human eyes have high-resolution vision only in the central 1-2 visual degrees of the visual field, known as the fovea, covering only the width of a finger at arm’s length. To compensate for the limited area of high visual acuity, humans learn to move their eyes to direct the foveae to the correct place at the right time to process important task-relevant visual information. In complex tasks, human eye movements are used by the visual system to a) identify structures in the environment that are critical for solving the task and b) exploit those structures by moving the high-resolution part of the visual field (fovea) to those locations via eye movements. Considerable evidence has shown that human gaze can be considered as an overt behavioral signal that encodes a wealth of information about both the motivation behind an action and the anticipated reward of an action [116, 117, 155]. The first research question is thus whether we can leverage recent progress in machine learning especially deep learning to model human visual attention given the gaze data.

Unlike human eyes, machines use cameras that have full resolution at every pixel. Despite this uniform visual acuity, for a given task, not all visual information is relevant at the current time and place. Redundant information poses a heavy burden on computational resources for AI systems. Visual perception is a key challenge in modern RL and IL research due to a high-dimensional state space (e.g., raw images) as input. Attention – the ability to identify, process, and respond to a reduced set of important features of visual input – thus could be also useful for RL and IL agents. Recent work has also proposed learning visual attention models from human gaze as an intermediate step towards learning the decision policy, and this intermediate signal has been shown to improve policy learning [187, 359, 335, 62, 191, 76].

We conjecture that machines could learn an attention mechanism from humans to discard irrelevant information and prioritize resources for the current task, and that integrating this mechanism into the machines will make them a lot more efficient. The second research goal is then to test this conjecture.

### **3.1 Summary of Contributions**

1. We introduce a large-scale dataset of human game playing behaviors: raw game image frames, player eye movements, actions (keyboard strokes), reaction time, performance (score), and many other behavioral measurements. The dataset has been made publicly available to encourage research in gaze modeling, as well as imitation learning and reinforcement learning (Section 3.2).

2. The gaze prediction problem is formulated as a saliency prediction problem. A convolution-deconvolution network can successfully predict human gaze positions (as a probability distribution), given raw game frames as input (3.3).
3. Predicting human decisions is formalized as an imitation through a supervised learning problem, in which the model learns to predict human action given a state (3.4).
4. A deep network (trained with imitation learning) can better predict human actions if aided by the learned human attention model. Knowing where a human allocates his or her attention improves action prediction accuracy (3.4).
5. The network with gaze information also performs better when playing the game, implying that human attention information is useful in guiding decision learning for artificial agents (3.4, 3.5).
6. We discuss how human attention is different from reinforcement learning agent's attention (3.6), and how to leverage human attention information to improve mainstream reinforcement learning algorithms (3.7).

## 3.2 Atari-HEAD Dataset<sup>1</sup>

### 3.2.1 Motivation

In modern cognitive science and machine learning, large-scale human datasets such as ImageNet [75] have played an important role in driving research progress. These datasets provide standardized benchmarks that ensure a fair comparison between models and algorithms.

In our context, imitation learning (2.3.1 results are difficult to reproduce since researchers often collect their own human data. During this process, many factors are uncontrolled – such as individual expertise, experimental setup, data collection tools, dataset size, and experimenter bias. A publicly available dataset would greatly reduce data collection efforts and allow algorithms to be compared with the same standard. Another concern with IL is the quality of the demonstration data. For supervised IL approaches like behavioral cloning, learning from sub-optimal demonstration can result in poor performance. Therefore the quality of human demonstration must be ensured.

Addressing the demands and challenges described above, we collected a large-scale dataset of humans playing Atari video games – one of the most widely used task domains in RL and IL research. The dataset is named Atari-

---

<sup>1</sup>This section of work is based on the following publication: Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl Muller, Jake Whritner, Luxin Zhang, Mary Hayhoe, and Dana Ballard. Atari-head: Atari human eye-tracking and demonstration dataset. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 6811–6820, 2020. The dissertator is the first author, and takes the leading role in conceiving and designing the analysis, collecting the data, contributing analysis tools, performing the analysis, and writing the paper.

HEAD (**A**tari **H**uman **E**ye-Tracking **A**nd **D**emonstration)<sup>2</sup>. An overview of this project can be found in Fig. 3.1. In collecting Atari-HEAD, we strictly follow standard data collection protocols for human studies and designed a special method to ensure the quality of demonstration policies. The result of these efforts is a dataset with expert-level task performance and minimal recording error. Making this dataset publicly available saves the effort of data collection and provides a benchmark for researchers who use Atari games as their task domain. Having both action and gaze data enables research that aims at bridging attention and control.

### 3.2.2 Task Domains and Data Collection

Games are ideal as a starting point since they are simple but capture many features of human cognition. Playing video games requires rapidly moving the eyes to perceive relevant information in a dynamic environment, memorizing important objects' locations, predicting the future state of the moving targets, and making decisions that maximize the expected reward. These cognitive abilities are fundamental for artificial intelligence, but only recently have AI programs been able to play simple Atari games with the skill of a human player [209].

Our data was collected using the Arcade Learning Environment (ALE) [30]. These games capture many interesting aspects of the natural visuomotor tasks while allowing better experimental control than real-world tasks. ALE

---

<sup>2</sup>Available at: <https://zenodo.org/record/3451402>

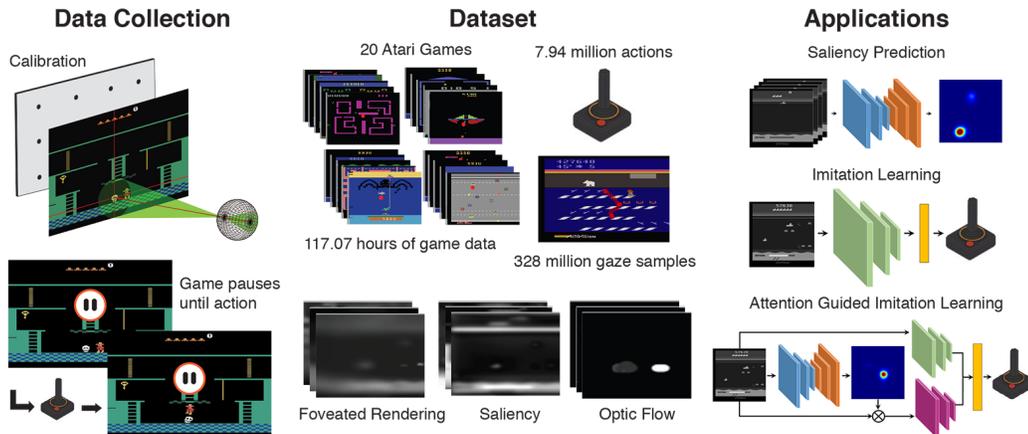


Figure 3.1: Project schematic for the Atari-HEAD dataset.

is deterministic given the same game seed. While collecting human data, the seed is randomly generated to introduce stochasticity for gameplay. We pick 20 games that span a variety of dynamics, visual features, reward mechanisms, and difficulty levels for both humans and AIs. An overview of this dataset can be found in Fig. 3.1. Game images along with eye-tracking data can be found in Appendix A.1.

For every game image frame  $i$ , we recorded its corresponding image frame  $I_i$ , human keystroke action  $a_i$ , human decision time  $t_i$ , gaze positions  $g_{i1} \dots g_{in}$ , and the immediate reward  $r_i$  returned by the environment. The gaze data was recorded using an EyeLink 1000 eye tracker at 1000Hz. The game screen was  $64.6 \times 40.0$ cm (or  $1280 \times 840$  in pixels), and the distance to the subjects' eyes was 78.7cm. The visual angle of an object is a measure of the size of the object's image on the retina. The visual angle of the screen was  $44.6 \times 28.5$  visual degrees. The human subjects were amateur players

who were familiar with the games. The human research was approved by the University of Texas at Austin Institutional Review Board with approval number 2006-06-0085. We collected data from 4 subjects playing 20 games. The total collected game time is 117.07 hours, with 7,937,159 action demonstrations and 328,870,044 usable gaze samples.

The subjects were only allowed to play for 15 minutes, and were required to rest for at least 15 minutes before the next trial. We mainly collected human data from the first 15 minutes of gameplay, since for most games AIs have not reached human performance at a 15-minute cutoff. Therefore we reset the game to start from the beginning for every trial. However, it is also interesting to know the human performance limit, hence for each game we let one human player play until the game terminated, or a 2-hour maximum time limit has been reached.

**Eye-tracking accuracy** The Eyelink 1000 tracker was calibrated using a 16-point calibration procedure at the beginning of each trial, and the same 16 points were used at the end of the trial to estimate the gaze positional error. The average end-of-trial gaze positional error across 471 trials was 0.40 visual degrees (or 2.94pixels/0.56cm), less than 1% of the stimulus size. Such high tracking accuracy is critical for Atari games, since many task-relevant objects are small.

### 3.2.3 Semi-Frame-by-Frame Game Mode

In the default ALE setting, the game runs continuously at 60Hz, a speed that is very challenging even for expert human players. Previous studies have collected human data, or evaluated human performance at this speed [209, 323, 170, 125]. However, we argue that to build a dataset useful for algorithms such as IL, a slower speed should be used. An innovative feature of our setup is that the game pauses at every frame until a keyboard action is taken by the human player. If desired, the subjects can hold down a key and the game will run continuously at 20Hz, a speed that is reported to be comfortable for most players. The reasons for such a setup are as follows:

**Resolving state-action mismatch** Closed-loop human visuomotor reaction time  $\Delta t$  is around 250-300 milliseconds. Therefore, during continuous gameplay,  $s_t$  and  $a_t$  that are simultaneously recorded at time step  $t$  could be mismatched. Action  $a_t$  could be intended for a state  $s_{t-\Delta t}$  250-300ms ago. An example that illustrates this point is shown in Fig. 3.2. This effect causes a serious issue for supervised learning algorithms, since label  $a_t$  and input  $s_t$  are no longer matched. Frame-by-frame gameplay ensures that  $s_t$  and  $a_t$  are matched at every timestep.

**Maximizing human performance** Frame-by-frame mode makes gameplay more relaxing and reduces fatigue, which could normally result in blinking and would corrupt eye-tracking data. More importantly, this design reduces

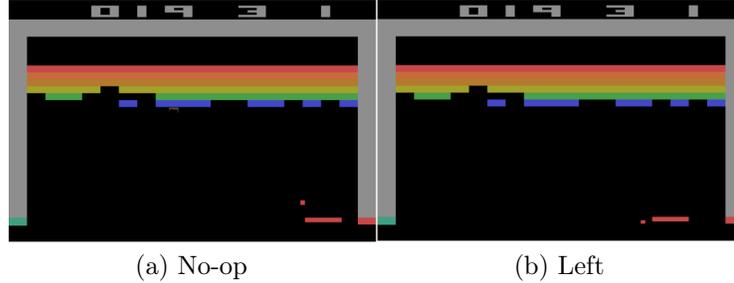


Figure 3.2: State-action mismatch (game Breakout, game speed at 60Hz). The state  $s_0$  at time  $t_0$  is shown in (a), the correct action would be to move the paddle left to catch the ball. However, due to the human player's delayed reaction, that action is executed 287ms (17 frames) later, as shown in (b). This delay leads to two undesirable consequences: 1) The player loses a life in the game; 2) Action "Left" is paired with state  $s_{17}$ , instead of  $s_0$ , which posits a serious issue for algorithms that attempts to learn the state-action mapping.

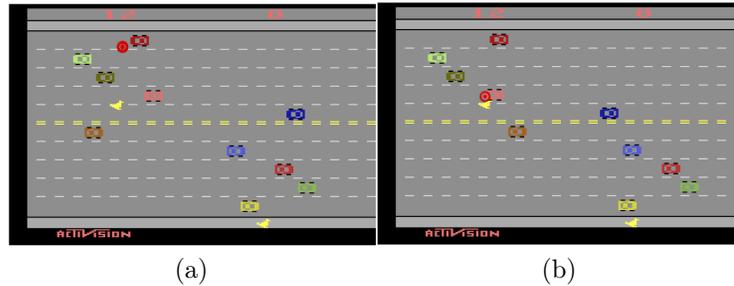


Figure 3.3: Inattentive blindness (game Freeway, game speed at 60Hz). (a) The player's attention (red dot) was on the red car. (b) The pink car hits the chicken controlled by the player 205ms later. Due to the fast pace of the game, the human player was not able to make an eye movement to attend and respond to the pink car.

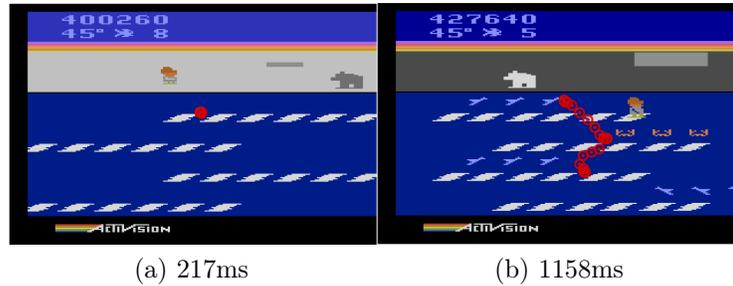


Figure 3.4: Scanpath and reaction time (game Frostbite, frame-by-frame mode). (a) A simple game state that only takes one fixation and 217ms to make a decision. (b) A complicated state which requires a sequence of eye movements and 1158ms to plan the next action. Our game mode allows enough time for the human player to process visual information and find the optimal action.

sub-optimal decisions caused by inattentive blindness. See Fig. 3.3 for an example.

**Highlighting critical states that require multiple eye movements** Human decision time and all eye movements were recorded at every frame. Hypothetically, the states that could lead to a large reward or penalty, or the ones that require sophisticated planning, will take longer and require multiple eye movements for the player to make a decision. Fig. 3.4 shows an example. Stopping gameplay means that the observer can use eye movements to resolve complex situations like (b). This is important because if the algorithm is going to learn from eye movements it must contain all “relevant” eye movements.

	Mnih	Wang	Hester	Kurin	de la Cruz	AtariHEAD 15-min avg.	AtariHEAD 15-min best	AtariHEAD 2-hour	Community Record	RL
alien	6,875	7,127.7	29,160	-	-	27,923	34,980	<b>107,140</b> <sup>†</sup>	103,583	9,491.7
asterix	8,503	8,503.3	18,100	-	14,300	110,133.3	135,000	<b>1,000,000</b> <sup>‡</sup>	<b>1,000,000</b>	428,200.3
bank_heist	734.4	753.1	7,465	-	-	5,631.3	6,503	<b>66,531</b> <sup>†</sup>	47,047	1,611.9
berzerk	-	2,630.4	-	-	-	6,799	7,950	55,220*	<b>171,770</b>	2,545.6
breakout	31.8	30.5	79	-	59	439.7	554	<b>864</b> <sup>‡</sup>	<b>864</b>	612.5
centipede	11,963	12,017	-	-	-	45,064	55,932	415,160*	<b>668,438</b>	9,015.5
demon_attack	3,401	3,442.8	6,190	-	-	7,097.3	10,460	107,045*	108,075	<b>111,185.2</b>
enduro	309.6	860.5	803	-	-	336.4	392	<b>4,886</b> *	-	2,259.3
freeway	29.6	29.6	32	-	-	31.1	33	33 <sup>‡</sup>	<b>34</b>	<b>34.0</b>
frostbite	4,335	4,334.7	-	-	-	31,731.5	50,630	<b>453,880</b> *	418,340	9,590.5
hero	25,763	30,826.4	99,320	-	-	59,999.8	77,185	541,640*	<b>1,000,000</b>	55,887.4
montezuma	4,367	4,753.3	34,900	27,900	-	38,715	46,000	270,400*	<b>400,000</b>	384.0
ms_pacman	15,693	15,375.0	55,021	29,311	18,241	28,031	36,061	93,721 <sup>†</sup>	<b>123,200</b>	6,283.5
name_this_game	4,076	8,049.0	19,380	-	4,840	7,661.5	8,870	<b>21,850</b> <sup>†</sup>	21,210	13,439.4
phoenix	-	7,242.6	-	-	-	30,800.5	40,780	<b>485,660</b> *	373,690	108,528.6
riverraid	13,513	17,118	39,710	-	-	20,048	22,590	59,420 <sup>†</sup>	<b>86,520</b>	-
road_runner	7,845	7,845	20,200	-	-	78,655	99,400	99,400 <sup>†</sup>	<b>210,200</b>	69,524.0
seaquest	20,182	42,054.7	101,120	-	-	52,774	64,710	<b>585,570</b> *	294,940	50,254.2
space_invaders	1,652	1,668.7	-	3,355	1,840	3,527	5,130	49,340*	<b>110,000</b>	18,789.0
venture	1,188	1,187.5	-	-	-	8,335	11,800	<b>28,600</b> <sup>†</sup>	-	1,107.0

Table 3.1: A comparison of human scores for 20 Atari games across datasets. The scores reported for [125, 170, 74] are the best human scores of each game. [209] and [323] are average scores. The community world record is from Twin Galaxies, an official supplier of verified world records by Guinness World Records. Note that the display and game difficulty may vary slightly across platforms, here we try to find the game version that matches our setting to the best of our knowledge. For Atari-HEAD 2-hour performance, <sup>†</sup>: game terminated. \*: Two-hour experiment time limit has been reached before the game terminated. If the human players continue to play, they could potentially achieve higher scores. <sup>‡</sup>: Maximum score allowed by the game reached.

### 3.2.4 Dataset Statistics

The experimental designs result in a high-quality human demonstration dataset. The optimality of demonstrated actions can be intuitively measured by final game scores (when the players lose their last life). In Table 3.1, we compare our human scores with ones reported in previous literature, along with Atari game world records, as well as one of the best RL agent’s performances [124]. We reported the average and the best game scores in 15-minute trials, as well as the highest score reached in the 2-hour gameplay mode. The immediate observation is that our design leads to better human performance compared to those previously reported. The community world record is from Twin Galaxies<sup>3</sup>, an official supplier of verified world records by Guinness World Records. For 8 games, our human players have obtained comparable or better scores than world records. For 6 other games, the 2-hour time limit was reached but the human players could surpass the world record if they continued to play.

In recent years, the gap between human and machine performance in many tasks has substantially narrowed [209]. AI agents such as DQN play the game in a frame-by-frame manner (although reaction time is not a big issue for RL agents), but in previous literature, humans played the game continuously at 60Hz. In our case, allowing human players to have enough decision time sets a stronger human performance baseline for RL agents. Our human score statistics indicate that humans retain advantages in these games, especially ones that

---

<sup>3</sup><https://www.twingalaxies.com/games>

require multitasking and attention. For difficult games recognized by the RL research community, such as Montezuma’s Revenge, human performance is still much higher than that of AI.

### 3.3 Gaze Modeling as Saliency Prediction<sup>4</sup>

Next, we will introduce the first modeling tasks that can be accomplished with the Atari-HEAD dataset: learning attention from humans. We will define the modeling task, discuss inputs and outputs of the models, propose evaluation metrics, and show baseline modeling results.

#### 3.3.1 Task Definition

The first learning task could be training an agent to imitate human gaze behaviors, i.e., learning to attend to important regions of a given image. The problem is formalized as a visual saliency prediction problem in computer vision research. The problem can be formulated as:

Given a state  $s_t$ , learn to predict human gaze positions  $g_t$ , i.e., learn  $P(g|s)$ .

---

<sup>4</sup>This section of work is based on the following publications: Ruohan Zhang, Zhuode Liu, Luxin Zhang, Jake A Whritner, Karl S Muller, Mary M Hayhoe, and Dana H Ballard. Agil: Learning attention from human for visuomotor tasks. In Proceedings of the European Conference on Computer Vision (ECCV), pages 663–679, 2018. Luxin Zhang, Ruohan Zhang, Zhuode Liu, Mary M Hayhoe, and Dana H Ballard. Learning attention model from human for visuomotor tasks. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018. The dissertator is the first/second author in these two papers, and takes the leading role in conceiving and designing the analysis, collecting the data, contributing analysis tools, performing the analysis, and writing the papers.

**Inputs and outputs** In the above formulation, note that  $g_t$  could be a set of positions in our dataset.  $s_t$  could be a single image  $I_t$ , or it could include a stack of images  $I_{t-n} \dots I_t$  to take into account more history. This includes information such as motion that can make states Markovian [209]. The images are published in RGB format, but it is common to convert them to be grayscale [209]. Note that for this dataset, two adjacent images, actions, or gaze locations are highly correlated. We suggest that users split data first then shuffle, instead of shuffle first then split, so one can avoid putting one frame in the training set and its neighboring frame in the testing set.

In saliency prediction, additional image statistics are shown to be correlated with visual attention and useful for gaze prediction [228, 187, 359]. We also provide tools to extract optical flow [87] and hand-crafted bottom-up saliency features (orientation and intensity) [144]. Examples of these can be seen in the third and fourth columns of Fig. 3.7. They can be directly used as reasonable guesses for gaze locations.

The gaze prediction model should output  $P(g_t|s_t)$ . In standard practice, discrete gaze positions are converted into a continuous distribution [52] by blurring each fixation location using a Gaussian with  $\sigma$  equals to one visual degree [176]. Hence the gaze prediction model will learn to predict this continuous probability distribution over the given image, which will be referred to as a saliency map.

**Evaluation metrics** Once the conversion is done, at least eight well-known metrics can be applied to measure prediction accuracy [52]. Let  $P$  denote the predicted saliency map,  $Q$  denote the ground truth, and  $i$  denote the  $i$ th pixel. We discuss four selected metrics here:

- **Area Under ROC Curve (AUC):** between 0 and 1. One can treat predicted saliency map as a binary classifier to indicate whether a pixel is fixated or not. Hence AUC, one of the most widely used metric in signal detection and classification problems can be applied here.
- **Normalized Scanpath Saliency (NSS):** This metric measures the normalized saliency at gaze positions by subtracting the mean predicted saliency value. It is sensitive to false positives and differences in saliency across predicted saliency map, but is invariant to linear transformations like contrast offsets:

$$NSS(P, Q) = \frac{1}{\sum_i Q_i} \sum_i \left( \frac{P_i - \mu(P)}{\sigma(P)} \times Q_i \right) \quad (3.1)$$

- **Kullback-Leibler Divergence (KL):** This metric is widely used to measure the difference between two probability distributions. It is also differentiable hence can be used as the loss function to train neural networks:

$$KL(P, Q) = \sum_i Q_i \log \left( \epsilon + \frac{Q_i}{\epsilon + P_i} \right) \quad (3.2)$$

$\epsilon$  is a small regularization constant and determines how much zero-valued predictions are penalized. KL is asymmetric and very sensitive to zero-valued predictions.

- **Pearson’s Correlation Coefficient (CC):** between 0 and 1. It measures the linear relationship between two distributions.

$$CC(P, Q) = \frac{\sigma(P, Q)}{\sigma(P) \times \sigma(Q)} \quad (3.3)$$

where  $\sigma(P, Q)$  denotes the covariance. CC is symmetric and penalizes false positives and negatives equally.

Note that KL and CC are distribution-based metrics, therefore the aforementioned process of converting discrete gaze positions to distributions is mandatory. However, for location-based metrics (AUC and NSS) the conversion is optional. Other usable metrics include Information Gain, Histogram Intersection, Shuffled AUC, Earth Mover’s Distance. For a comprehensive survey about their properties, please see [52].

### 3.3.2 Baseline Model and Results

We trained a convolution-deconvolution gaze network [228, 359, 351, 76] with KL divergence ( $\epsilon = 1e - 10$ ) as loss function to predict human gaze positions. The details of the network design can be found in Fig. 3.5 (more technical details are in Appendix A.2). A separate network is trained for each game. We use 80% data for training and 20% for testing.

Aggregated modeling results can be seen in Fig. 3.6. As expected, the learning-based neural network model outperforms optical flow and bottom-up saliency models by a large margin in all metrics. The prediction accuracy overall is high (average AUC of 0.971), although varies across games (min AUC:

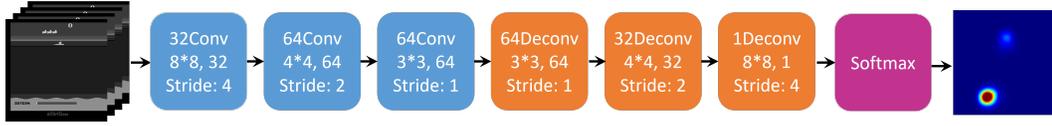


Figure 3.5: The gaze prediction network. The network takes in a stack of 4 consecutive game images in grayscale, passes the inputs to 3 convolutional layers followed by 3 deconvolutional layers. The final output is a gaze saliency map that indicates the predicted probability distribution of the gaze.

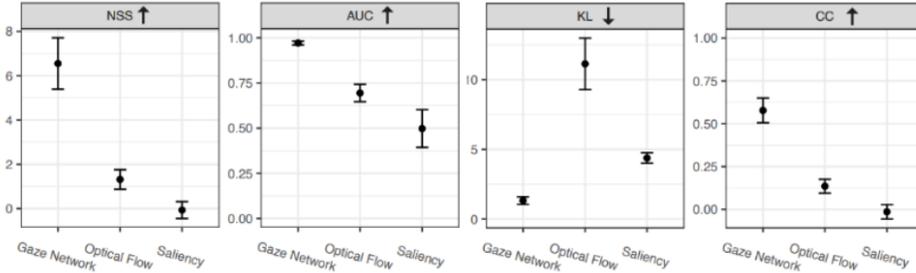
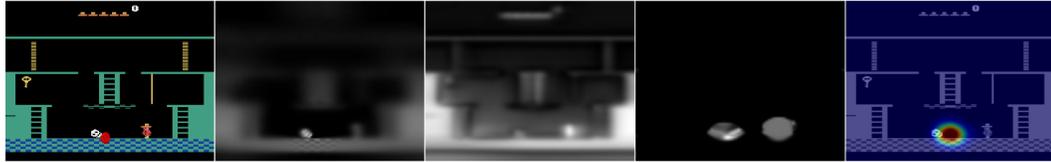


Figure 3.6: Gaze prediction results measured using four standard metrics, averaged across 20 games. As expected, a convolution-deconvolution network (gaze network) is able to predict human gaze much more accurately than motion-based and image saliency-based models. Error bars indicate standard deviation across games (N=20).

0.945-Ms.Pacman, max: 0.988-Enduro). Results for each individual game can be found in Appendix A.2.

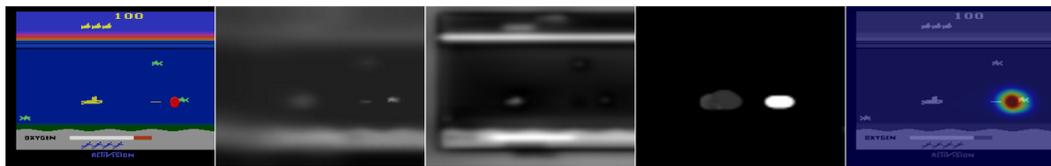
The predicted saliency maps can be visualized in Fig. 3.7. Additionally, we observe that the model predicts gaze behaviors that are difficult to capture using hand-defined features [364]. The model can capture predictive eye movements, as shown in Fig. 3.8a. Note that there are no salient visual features at the gaze location. The model can predict that the location is where the ball will be. The model can also tell visually identical task-relevant objects



(a) Montezuma's Revenge



(b) Ms. Pacman



(c) Seaquest



(d) Space Invaders

Figure 3.7: Gaze prediction results for four Atari games. First column: game screenshots with red dots indicating the human gaze positions. Second column: biologically plausible retinal image, generated by foveated rendering algorithm [237]. Third column: image saliency calculated by the classic Itti-Koch saliency model [144]. Fourth column: Farnebeck optical flow, calculated using the frame in the first column and its previous frame [87]. Fifth column: predicted gaze distribution by convolution-deconvolution network, overlaid on top of the original image.

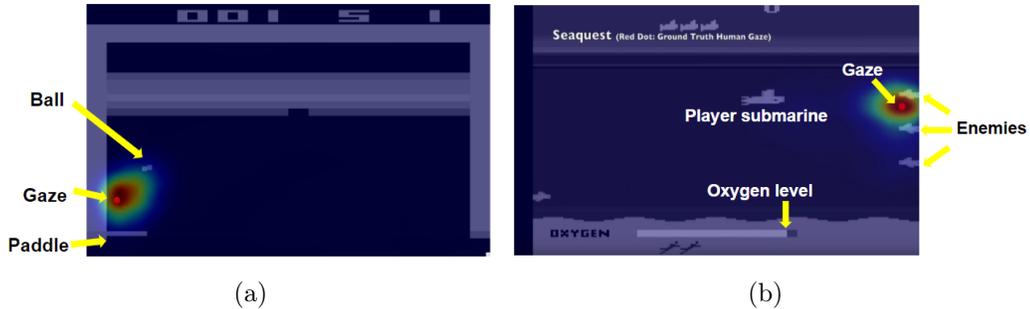


Figure 3.8: The gaze model predicts gaze behaviors that are difficult to capture using hand-defined features. (a) The model captures predictive eye movements in game Breakout. (b) The model can also tell visually identical task-relevant object for the moment from others in Seaquest.

for the moment from others, as shown in Fig. 3.8b. There are three visually identical enemies but only the top one presents an immediate threat to the player submarine.

The saliency prediction results using the dataset are considered highly accurate in saliency prediction research. One reason is the large amount of training data available provided by the dataset. Another reason is that the chosen tasks are reward-seeking and demanding, therefore human gaze is mostly directed towards image features that are strongly associated with reward and hence become highly predictable [116, 117].

### 3.3.3 Gaze Prediction with Additional Information

To further improve the prediction accuracy, especially with limited data, researchers can optionally use additional inputs (motion, bottom-up saliency, or image semantics) along with the original images to predict human gaze

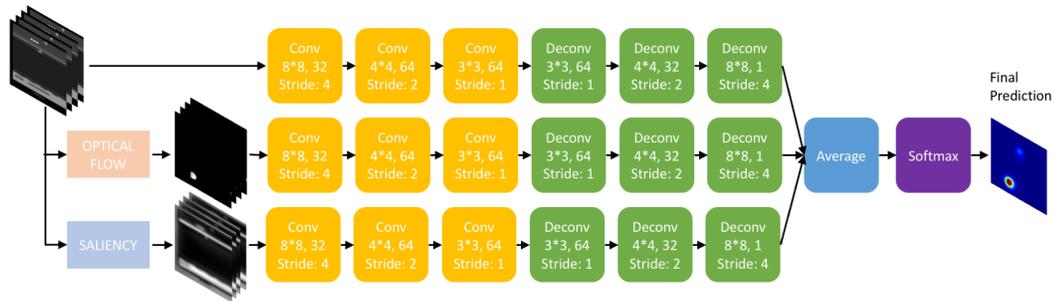


Figure 3.9: The three-channel gaze prediction network. The top channel takes in images, the mid channel takes in the corresponding optical flow, and the bottom channel takes in the bottom-up image saliency. We then average the output of three channels. The final output is a gaze saliency map that indicates the predicted probability distribution of the gaze. The design of the convolutional layers follows the Deep Q-network [209].

positions. Several previous works have shown these signals are helpful for gaze prediction in visuomotor tasks [359, 228, 187]. The network architecture we use (shown in Fig. 3.9) is a three-channel convolution-deconvolution network. The inputs to the top channel are the images where the preprocessing procedure follows [209] and hence consists of a sequence of 4 frames stacked together. The mid-channel models motion information (optical flow) which is included since human gaze is sensitive to movement. Optical flow vectors of two continuous frames are calculated using the algorithm in [87] and fed into the network. The bottom channel includes a bottom-up saliency map computed by the classic Itti-Koch model [144]. Instead of using the full dataset, we only use half of the data (8 trials per game, 120-minute) for training, and we only selected 8 games.

The quantitative results are shown in Table 3.2. A two-channel model

(Image+Motion) in general achieves the best results. Further removing the motion information (having only the image) results in only slightly less accuracy—except for the game Venture in which the speed of the monsters matters the most, hence removing motion decreases prediction accuracy. Including bottom-up saliency in the model does not improve the performance overall. This indicates that in the given tasks, the top-down visual attention is different than and hard to be inferred from the traditional bottom-up image saliency.

The main reason to include motion and saliency information is to improve sample efficiency with limited data. We study the effect of varying training sample size on prediction accuracy and find that the Image+Motion model can achieve high AUC values (above 0.88 for MsPacman and above 0.94 for 7 other games) with a single trial of human gaze data (15-minute)—although additional data can still help. The learning curves plotted against the training sample size for all games can be found in Appendix A.2. Therefore, training the gaze network does not incur a heavy burden on sample size for the given task.

However, motion or saliency information is contained in the current input state (a stack of four consecutive image frames). Therefore with enough training data, the network can learn to extract such information. Using the full dataset as training (16 trials per game), we found that including motion and/or saliency information does not significantly improve the gaze prediction accuracy, in contrast to the results showing above which only use limited training data [358, 359, 351].

		Break-out	Free-way	Enduro	River-raid	Sea-quest	Ms-Pacman	Centi-pede	Ven-ture
Saliency(S)	NSS↑	-0.075	-0.175	-0.261	0.094	-0.208	-0.376	0.665	0.422
Motion(M)		2.306	1.015	0.601	1.200	2.016	0.891	1.229	1.004
Image(I)		6.336	6.762	8.455	5.776	6.417	4.522	5.147	5.429
I+S		6.363	6.837	8.379	5.746	6.384	4.518	5.215	5.469
I+M		<b>6.432</b>	<b>6.874</b>	<b>8.481</b>	5.834	6.485	<b>4.600</b>	<b>5.445</b>	<b>6.222</b>
I+S+M		6.429	6.852	8.435	<b>5.873</b>	<b>6.510</b>	4.571	5.369	6.125
Saliency(S)	AUC↑	0.494	0.560	0.447	0.494	0.352	0.426	0.691	0.607
Motion(M)		0.664	0.697	0.742	0.738	0.779	0.664	0.729	0.643
Image(I)		<b>0.970</b>	<b>0.973</b>	<b>0.988</b>	<b>0.962</b>	0.963	0.932	0.956	0.957
I+S		0.969	<b>0.973</b>	<b>0.988</b>	0.961	0.963	0.933	0.957	0.956
I+M		<b>0.970</b>	0.972	<b>0.988</b>	<b>0.962</b>	<b>0.964</b>	0.935	<b>0.961</b>	<b>0.964</b>
I+S+M		0.969	<b>0.973</b>	<b>0.988</b>	<b>0.962</b>	<b>0.964</b>	<b>0.936</b>	0.960	<b>0.964</b>
Saliency(S)	KL↓	4.375	4.289	4.517	4.235	4.744	4.680	3.774	3.868
Motion(M)		13.097	10.638	8.312	9.151	9.133	12.173	10.810	12.853
Image(I)		1.304	1.261	0.834	1.609	1.464	1.985	1.711	1.749
I+S		1.301	1.260	0.834	1.613	1.470	1.995	1.709	1.727
I+M		<b>1.294</b>	<b>1.257</b>	<b>0.832</b>	1.593	1.438	<b>1.959</b>	<b>1.622</b>	1.512
I+S+M		1.299	1.260	0.835	<b>1.592</b>	<b>1.437</b>	1.961	1.645	<b>1.510</b>
Saliency(S)	CC↑	-0.009	-0.023	-0.033	-0.008	-0.035	-0.048	0.065	0.048
Motion(M)		0.205	0.099	0.077	0.125	0.190	0.092	0.132	0.105
Image(I)		0.583	0.588	0.705	0.505	0.558	0.439	0.481	0.483
I+S		0.583	0.588	0.702	0.503	0.555	0.436	0.479	0.488
I+M		<b>0.584</b>	<b>0.591</b>	<b>0.706</b>	0.509	<b>0.564</b>	<b>0.441</b>	<b>0.499</b>	<b>0.543</b>
I+S+M		<b>0.584</b>	0.589	0.704	<b>0.511</b>	0.562	0.440	0.492	0.541

Table 3.2: Quantitative results of predicting human gaze across eight games. Random prediction baseline: NSS = 0.000, AUC = 0.500, KL = 6.159, CC = 0.000. For comparison, the classic [144] model (Saliency) and optical flow (Motion) are compared to versions of our model. All our models are accurate in predicting human gaze (AUC>0.93). In general the Image+Motion (I+M) model achieves the best prediction accuracy across games and four metrics.

### 3.3.4 Individual Gaze Differences

Do human subjects exhibit different gaze behaviors when performing the same task? This question is further investigated by training the gaze network on one subject’s data and testing on the others for all games. We find that the gaze model is most accurate when trained and tested on the same subject. When tested on a different subject, the average prediction accuracy loss, in terms of the correlation coefficient, is 0.091 compared to trained and tested on the same subject (0.387 vs. 0.478). A detailed analysis can be found in Appendix A.2.

## 3.4 Decision Modeling with Attention Information<sup>5</sup>

### 3.4.1 Task Definition

The next task is to learn from human demonstrated actions. Recall the standard imitation learning problem is formulated as (Section 2.3.1):

Given a state  $s_t$ , learn to predict human action  $a_t$ , i.e., learn  $P(a|s)$ ,  
or equivalently, policy  $\pi(s, a)$ .

With human gaze data, we propose to use attention information to improve policy learning. This attention-guided learning problem is formulated as follows:

---

<sup>5</sup>This section of work is based on the following publication: Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl Muller, Jake Whritner, Luxin Zhang, Mary Hayhoe, and Dana Ballard. Atari-head: Atari human eye-tracking and demonstration dataset. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 6811–6820, 2020. The dissertator is the first author, and takes the leading role in conceiving and designing the analysis, collecting the data, contributing analysis tools, performing the analysis, and writing the paper.

Given a state  $s_t$  and human gaze positions  $g_t$ , learn to predict human action  $a_t$ , i.e., learn  $P(a|s, g)$ .

Another potential formulation is a joint learning problem:

Given a state  $s_t$ , learn to jointly predict human action  $a_t$  and gaze positions  $g_t$ , i.e., learn  $P(a, g|s)$ .

### Evaluation metrics

- **Behavior matching accuracy:** It measures the accuracy in predicting human actions.
- **Game score:** The model that predicts human actions is effectively a gaming AI. Its performance can be directly measured by the final game score.

Note that the results on these two metrics may not necessarily be correlated, as we will show later.

## 3.4.2 Models and Results

### 3.4.2.1 Behavioral Cloning

For standard behavioral cloning (Section 2.3.1.1), since Atari games have a discrete action space (18 actions), one can treat the prediction task as an 18-way classification problem with standard log likelihood loss:

$$J = - \sum_t^T \sum_{a=0}^{17} \mathbb{1}_{a_t=a} \log P(a_t = a|s_t) \quad (3.4)$$



Figure 3.10: A behavioral cloning network for predicting human actions. The network takes in a single grayscale game image as input, and outputs a vector that gives the probability of each action. We can replace the input image with its foveated version.

we trained a convolutional network, shown in Fig. 3.10 using the classification loss above.

### 3.4.2.2 Behavioral Cloning with Biologically Plausible State Representation<sup>6</sup>

Although the environment presents the same visual stimulus to humans and the machine, the underlying state representations may not be the same, partially due to the differences in perceptual systems. This leads to discrepancy in perceived states of human and machine, where the machine perceives images like in Figs. 3.11a while a human may see Figs. 3.11c-3.11d. In [357] we hypothesize that training the network with realistic retinal images may improve prediction, since these images are closer to the true human representation. We fed the visual angle of the game screen (45 degrees), gaze positions, and images into the Space Variant Imaging system [237]<sup>7</sup>. The software provides

<sup>6</sup>This section of work is based on the following publication: Ruohan Zhang, Zhuode Liu, Mary M Hayhoe, and Dana H Ballard. Attention guided deep imitation learning. In Cognitive Computational Neuroscience (CCN), 2017. The dissertator is the first author, and takes the leading role in conceiving and designing the analysis, collecting the data, contributing analysis tools, performing the analysis, and writing the paper.

<sup>7</sup><http://www.cps.utexas.edu/svi/>

a biologically plausible simulation of the foveated retinal images as shown in Figs. 3.11b. The foveated images are fed into the behavioral cloning network. We use 15-minute human data from the game Seaquest.

We vary the visual angle to change the clarity of the images as if the subjects were viewing the game screen from various distances, as shown in Figs. 3.11c-3.11d. The actual visual angle is 45 degrees in our experiments. We observe that the prediction accuracy decreases when deviated from the true value, as shown in Fig. 3.12, but they all outperform the baseline without gaze information (41.44%) significantly.

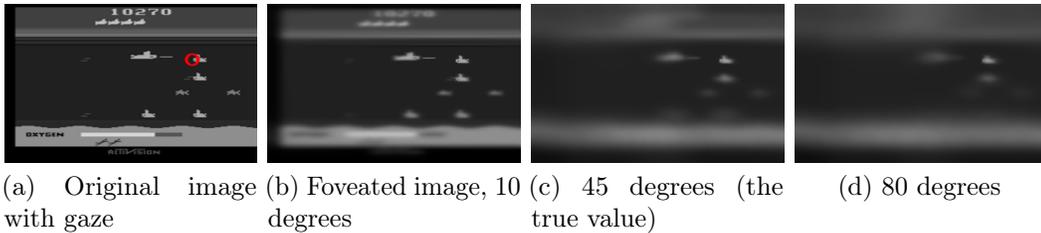


Figure 3.11: The original game frames for Atari Seaquest with red circles indicating the gaze position. The gaze position is used to generate the foveated images. Visual degree indicates the size of the game screen in the visual field.

### 3.4.2.3 Attention-Guided Imitation Learning (AGIL)

The experiment above demonstrates the potential benefit of using human gaze information. However, the human gaze data would not be available if we want the agent to perform the task on its own, and generating foveated images for every frame is very computationally expensive. To better incorporate gaze information into IL, we design a two-channel policy network [359]. The policy

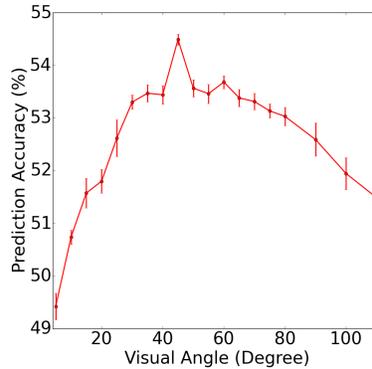


Figure 3.12: Action prediction accuracy for models trained using foveated images with different visual angles. 45 degrees is the correct visual angle that yields the best performance.

network uses the saliency map predicted by the gaze network to mask the input image. This mask can be applied to the image to generate a “foveated” representation of the image that highlights the attended visual features [187, 357, 359, 335, 62]. The design of the network (AGIL) can be found in Fig. 3.13.

The performance measured by behavior matching accuracy can be seen in Fig. 3.14 (detailed results are in Appendix A.3), compared against the behavioral cloning network without attention information. The main result is that incorporating attention improves accuracy on all games with an average improvement of 7%. However, the magnitude of improvement varies across games. The games with most improvements are Name This Game (19%), Alien (19%), Seaquest (16%), Ms.Pacman (12%), Asterix (12%), and Frostbite (12%). These are games where many task-relevant objects appear on the screen simultaneously. As a result, the current behavioral target is often ambiguous without attention information, therefore incorporating attention leads to better

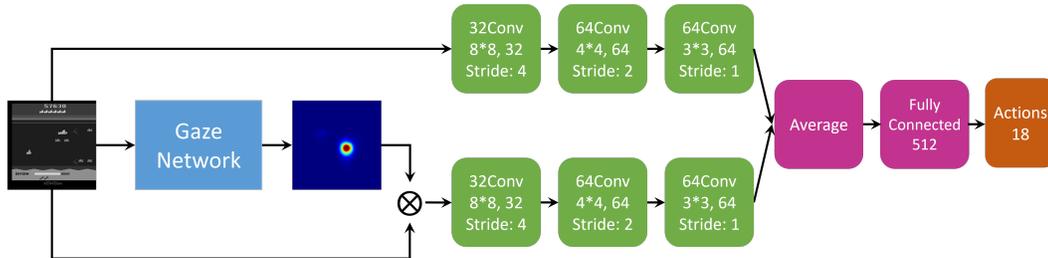


Figure 3.13: The AGIL policy network architecture for predicting human actions. The top channel takes in the current image frame and the bottom channel takes in the masked image which is an element-wise product of the original image and predicted gaze saliency map by the gaze network. We then average the output of the two channels.

prediction.

We look at game scores obtained by different models using different datasets, shown in Fig. 3.15 (detailed results are in Appendix A.3). We include IL (behavior cloning) results from two previous datasets [125, 170]. Applying IL and AGIL [359] to our dataset, the mean scores are averaged over 500 episodes per game, with each episode initialized with a randomly generated seed. The game is cut off after 108K frames [124]. The agent chooses an action  $a$  probabilistically using a softmax function with Gibbs (Boltzmann) distribution according to the policy network’s prediction  $P(a)$ :  $\pi(a) = \frac{\exp(\eta P(a))}{\sum_{a' \in \mathcal{A}} \exp(\eta P(a'))}$  where  $\mathcal{A}$  denotes the set of all possible actions,  $\exp(\cdot)$  denotes the exponential function, and the temperature parameter  $\eta$  is set to 1. The scale and quality of our data lead to better performance when comparing to AtariHEAD-IL to Kurin-IL and Hester-IL. The AtariHead-AGIL agent first learns to predict human gaze and uses the learned gaze model to guide the process of learning human decisions. This result shows that attention information is useful for IL.

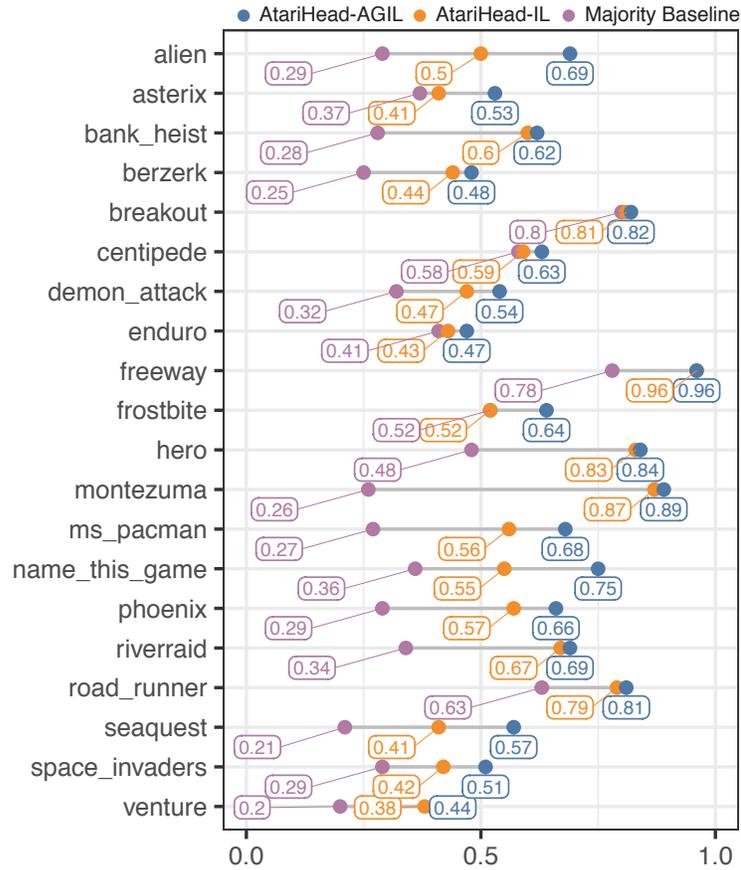


Figure 3.14: Incorporating gaze model learned from humans improves imitation learning algorithm’s performance in terms of behavior matching accuracy. The majority baseline simply predicts the majority class in that game (the most frequent action). IL: Standard behavioral cloning. AGIL: policy network that includes saliency map predicted by the gaze network. Random guess accuracy: 0.06.

The AGIL model improves game performance on 19 games, with an average improvement of 115.26%.

Note that the results on the accuracy and score metrics may not necessarily be strongly correlated. For instance, a 1% increase in accuracy leads to a 1138% improvement in scores for the game Breakout, while for Space Invaders, a 9% increase leads to minor improvement (0.45%) in game scores.

### 3.4.3 Why Attention Helps

Why does the learned visual attention model improve action prediction accuracy and task performance [364]? First, attention highlights task-relevant visual features in a high-dimensional state space, even though the features may only occupy a few pixels in that space, as observed in Figs 3.7. Hence, attention can be seen as a feature selection mechanism that biases the policy network to focus on the selected features. Second, attention could help to identify and disambiguate the goal of current action when multiple task-relevant objects are present. For example, in Fig. 3.16b and 3.16c, the gaze indicates that the goal of the current action involves the enemy to the left or above the yellow submarine. The corresponding actions would be moving left for the first case and moving up for the second. The two enemies are visually identical hence the learning agent cannot predict the correct action without gaze information – an issue further exacerbated by convolutional layers of the network due to their spatial invariant nature. For these reasons, modeling human attention helps the agent infer the correct decision state of the human teacher and understand

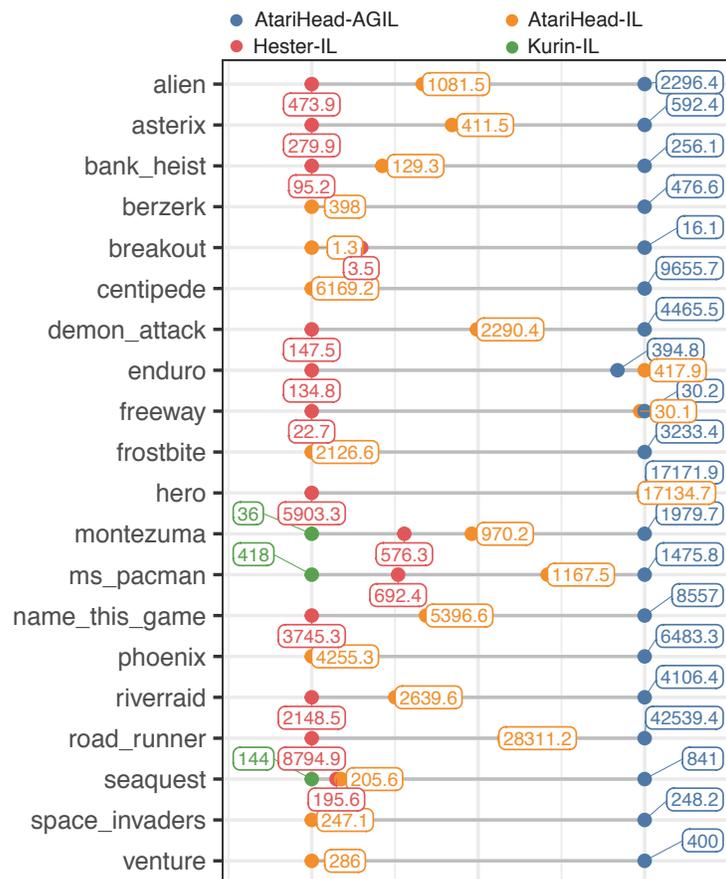


Figure 3.15: Incorporating gaze model learned from humans improves imitation learning algorithm’s performance in terms of game scores. With the large-scale high-quality dataset we collected, an IL agent can perform better than similar agents reported in previous datasets [125, 170]. Additionally, incorporating the attention model learned from human gaze improves IL agent’s performance with an average improvement of 115.26%.



Figure 3.16: Human gaze information helps the learning agent correctly infer the underlying reason for the chosen action. The red circles indicate human teacher’s gaze position.

the underlying reason for that decision.

### 3.5 Extending AGIL: Coverage-Based Gaze Loss<sup>8</sup>

Section 3.3 and 3.4 mainly focused on modeling human gaze and decision behaviors. Meanwhile, we have shown that human attention information could directly improve a behavioral cloning agent’s task performance. If we focus on improving the performance of artificial learning agents, can we find a more suitable way to utilize human attention for these agents? Can we do better than AGIL and use human attention for other imitation learning algorithms?

Prior approaches such as AGIL utilizing gaze for IL algorithms use gaze heat maps as input (e.g., as a mask) to the agent’s learning model in addition to the world state [359, 191]. These approaches have four major drawbacks:

<sup>8</sup>This section of work is based on the following publication: Akanksha Saran, Ruohan Zhang, Elaine Schaertl Short, and Scott Niekum. Efficiently guiding imitation learning agents with human gaze. arXiv preprint arXiv:2002.12500, 2020. The dissertator is the second author, and contribute in conceiving and designing the analysis, collecting the data, contributing analysis tools, performing the analysis, and writing the paper.

1. These approaches increase model complexity. For example, AGIL (Fig. 3.13) doubles the number of trainable parameters when adding the second channel. Part of the performance improvement may come from these additional parameters instead of attention information.
2. They require gaze information at test time when the agent performs the task on its own, hence a gaze prediction model is needed.
3. It is unknown whether human gaze information can also benefit other imitation learning algorithms beyond behavioral cloning.
4. Human gaze only reveals their *overt* attention. However, humans can still pay *covert* attention to entities in the working memory [239]. Hence being attended by the human gaze model is a sufficient (but not necessary) condition for the features to be important. Most RL/IL agents do not have working memory systems thus may need additional features beyond those captured by human overt attention. Direct masking approaches discard those features.

To address these drawbacks, we propose using an auxiliary gaze loss, called coverage-based gaze loss (CGL) [268], during the training of imitation learning algorithms to improve the performance of existing methods without increasing model complexity, or requiring test-time gaze, meanwhile addressing the covert attention issue. In addition to behavioral cloning (BC), we use gaze cues to enhance the performance of other two IL algorithms — behavioral

cloning from observation (BCO) (Section 2.3.2.2), and an IRL algorithm (Section 2.3.1.2) called Trajectory-ranked Reward EXtrapolation (T-REX).

We find that our proposed approach improves the performance by 86.2% for BC, 343.6% for BCO, and 373.6% for T-REX, compared to models without attention information and averaged over 20 Atari games. We also find that compared to AGIL, our method achieves better performance, and is more efficient in terms of learning with fewer demonstrations. At last, we show that CGL can help alleviate a well-known causal confusion problem in imitation learning.

### 3.5.1 Method

We use an auxiliary gaze loss to guide the learning of any agent using image-based state representations and convolutional layers as part of its model architecture. Using gaze information as an auxiliary loss term addresses three issues we discussed above: (1) It only affects the features learned by existing deep networks and does not increase their model complexity by adding more parameters. (2) It is only used during training hence eliminates the need for gaze information at test time. (3) It can guide any convolutional network to attend towards features that human demonstrators attend to hence can be directly applied to any deep IL algorithm.

The remaining issue is the overt vs. covert attention one. Human gaze data only reveals overt attention which is directly connected to a sensory organ. However, humans can still pay covert attention to entities in the working

memory [239]. In other words, being attended by the human gaze model is a sufficient (but not necessary) condition for the features to be important. Our loss term will have a higher penalty if the network does not attend to parts of the image that human focused on, but will have *no* penalty for activations where the human did not pay attention. We refer to the proposed loss function as a coverage-based gaze loss (CGL).

CGL operates on the human gaze heatmap and the output of the last convolutional layer of a deep network. Activation feature maps from the last convolutional layer [272] of image classification CNNs are shown to have the best compromise between high-level semantics and detailed spatial information. Given a 3D feature map of size  $h \times w \times c$  from a convolutional layer, it is collapsed to a feature map  $f$  of size  $h \times w$  using a  $1 \times 1$  convolutional filter. Equation 3.6 shows the normalization of this feature map  $f$  using a softmax operator to values between 0 and 1. Given a normalized 2D gaze heatmap  $g$  of size  $h \times w$ , CGL is computed as:

$$CGL(g, f') = \sum_{i \in (1, h)} \sum_{j \in (1, w)} g_{i,j} \left[ \log \frac{g_{i,j} + \epsilon}{f'_{i,j} + \epsilon} \right] \quad (3.5)$$

where

$$f'_{i,j} = \frac{e^{f_{i,j}}}{\sum_{k=0}^{k=h-1} \sum_{j=0}^{j=w-1} e^{f_{k,l}}} \quad (3.6)$$

CGL is similar to KL divergence (Eq.3.2). It adds a penalty if activations from none of the convolutional filters are high on areas where the human gaze fixates during gameplay. If activations are non-zero in other areas where the human

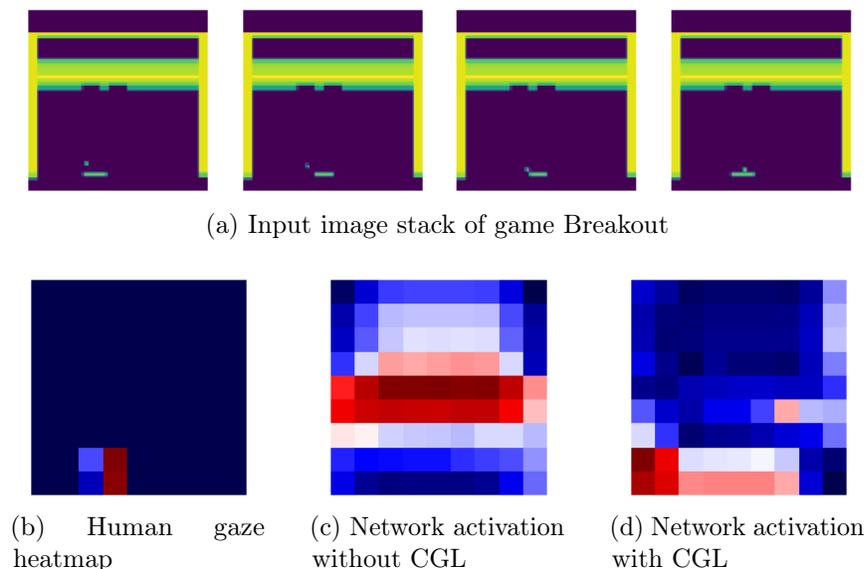


Figure 3.17: Our auxiliary gaze loss (CGL) guides a convolutional network to focus on parts of the state space which the human attends to (b). An example of network activations on the Breakout game (a) is shown in (c) without utilizing gaze and (d) when CGL is incorporated as part of the training. We find that when CGL is utilized, network activations are more heavily focused (in red) on the bottom of the input image stack (a), where the human also paid attention during demonstrations (b). Without CGL, the network chooses to focus more on the middle rows of the input image stack.

does not focus, there is no penalty. Hence, CGL encourages *coverage* of the human attended regions. The magnitude of the penalty is computed using a smoothed ( $\epsilon = 2.2204E^{-16}$ ) KL divergence term between the normalized gaze map and the collapsed and normalized convolutional map, and is then weighted by the amount of gaze fixation an image region gets. The effect of CGL on a trained CNN is visualized in Fig. 3.17.

### 3.5.1.1 Auxiliary Gaze Loss

For the behavioral cloning (BC) method, the gaze coverage loss is added as an auxiliary loss term in addition to the log likelihood action classification loss:

$$\mathcal{L}(\theta) = \sum_{i=1}^N \left[ - (1 - \alpha) \log \pi_{\theta}(a_i | s_i) + \alpha CGL(g(s_i), c_3(s_i)) \right] \quad (3.7)$$

The network architecture is similar to the behavioral cloning network in Fig. 3.10, comprising of three convolutional layers and one fully connected layer and taking a single frame as input. We decrease stride length at the first convolution layer to increase the size of the produced feature maps at the third convolution layer (from  $7 \times 7$  to  $21 \times 21$ ). The gaze coverage loss is applied to the feature maps at the third convolutional layer.  $g(s_i)$  is the gaze map of size  $21 \times 21$ ,  $c_3(s_i)$  is the collapsed and normalized feature map of size  $21 \times 21$  (Eq. 3.6) from the third convolutional layer, and  $N = 50$  is the batch size.

For behavioral cloning from observation (Section 2.3.2.2 [308], and T-REX [44] which is deep inverse reinforcement learning algorithm (Section 2.3.1.2, we incorporate CGL in their original loss functions in a similar way.

### 3.5.2 Other Techniques to Incorporate Gaze

Here we describe two alternative methods incorporating human gaze for imitation learning, which we compare against.

**Attention Guided Imitation Learning (AGIL)** AGIL (Section 3.4 adds more parameters to a BC network to utilize gaze. The output of the gaze

prediction network is used as input to an additional convolutional pathway in a modified version of standard behavioral cloning. AGIL consists of two channels of 3 convolutional layers. One channel takes as input a single image frame (game state) and another uses a masked image which is an element-wise product of the original image and predicted gaze saliency map. To ensure a fair comparison with CGL+BC, we decrease stride length at the first convolution layer to increase the size of the produced feature maps at the third convolution layer (from  $7 \times 7$  to  $21 \times 21$ ). The rest of the hyperparameters stay the same.

**Gaze-modulated Dropout (GMD)** As another baseline for learning from human gaze, we implement GMD [62] after the first two convolutional layers of the BCO policy network [308]. Gaze maps are generated using the gaze prediction network (Fig. 3.5).

### 3.5.3 Experiments and Results

We use data from all 20 games in Atari-HEAD. All reported results were game scores averaged over 30 different rollouts (episodes) of the learned policy. We used the default settings from OpenAI baselines [77] for parameters of ALE [31]. All experiments are conducted on server clusters with NVIDIA 1080, 1080Ti, Titan V, or DGX GPUs.

For evaluation, we intend to show improvement in terms of game scores using CGL. We calculate the improvement factor over baseline in the following way:  $\text{improvement} = (\text{new score} - \text{baseline score}) / \text{baseline score}$ . If both

the baseline score and the new score are zero, improvement is zero. However, for some games baseline game scores are zeroes but new scores are non-zero. In such cases, the improvement will not be calculated. We report average improvement (including games in which improvements are negative) across 20 games. Details on the scores of the individual games can be found in Appendix A.4. Note that the improvement factors are *underestimated*, due to the way we handle zero score games.

### 3.5.3.1 CGL Improves Behavioral Cloning

BC+CGL outperforms basic BC on 19 out of 20 games with 15-minutes of human demonstration data (Fig. 3.18). On average, the improvement is **95.1%**. With all 300 minutes of human gameplay data, BC+CGL outperforms BC on all 20 games with an average improvement of **86.2%**. The hyperparameter  $\alpha$  is tuned using a grid search from a set of 7 values — 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9. The Adadelta optimizer [346] is used for all models based on BC.

**Efficiency of CGL in terms of learnable parameters** Previous methods (such as AGIL) incorporate human attention by introducing extra parameters to the model due to additional neural network modules added. To tease apart whether improvement in these approaches comes from increased parameters to standard behavioral cloning or from the gaze information itself, we perform the following experiment. We re-train the AGIL network, but instead of using gaze heatmaps, we pass the original image as input to the gaze pathway, referred

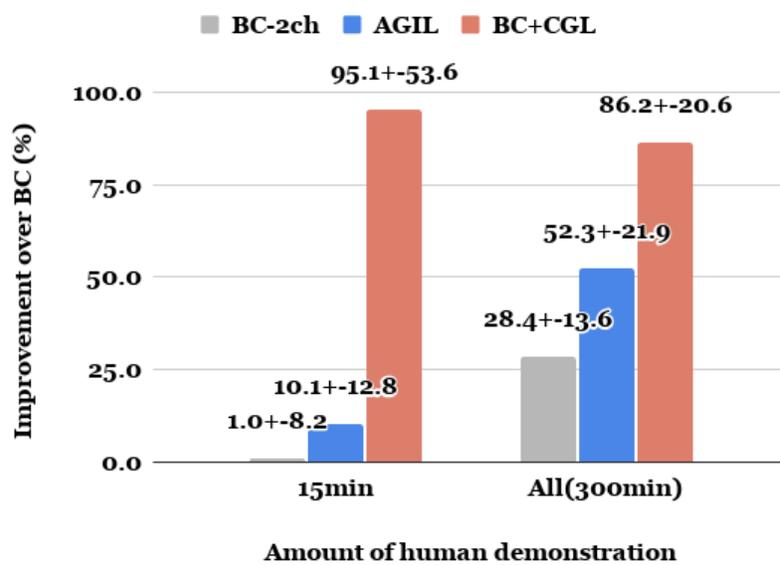


Figure 3.18: Average (across 20 games) percentage improvement over the BC baseline. Results are presented as as mean±standard error of the mean (N=20). Individual game scores of all agents can be found in Appendix A.4.

to as the BC-2ch model. This helps us disambiguate if more parameters in the model help more versus the gaze information itself. As shown in Fig. 3.18, we find that the BC-2ch model does result in improved performance over BC, indicating that part of AGIL’s improvement over BC is due to additional parameters. This hints at the fact that increasing model complexity alone without using any additional information as input proves beneficial.

**CGL provides stronger guidance than AGIL** In Fig. 3.18, we also show that on average, CGL outperforms the previously best method to incorporate gaze (AGIL [359]) by a large margin. This study suggests that CGL performs significantly better than AGIL with fewer model parameters, and the advantage is even more evident with a limited amount of human demonstration data (95.1% improvement over BC versus 10.1%). The sample efficiency of CGL is critical as it can be beneficial for applying this method to challenging imitation learning problems, where collecting demonstrations is cumbersome and expensive.

### 3.5.3.2 CGL Improves Behavioral Cloning from Observation

BCO+CGL outperforms basic BCO on 14 out of 20 games with 15-minute human demonstration data (Fig. 3.19). On average, the improvement is **160.9%**. With all 300-minute human data, BCO+CGL outperforms BCO on 12 out of 20 games with an average improvement of **343.6%**. We found that BCO is unable to learn an accurate inverse dynamics model for up to six

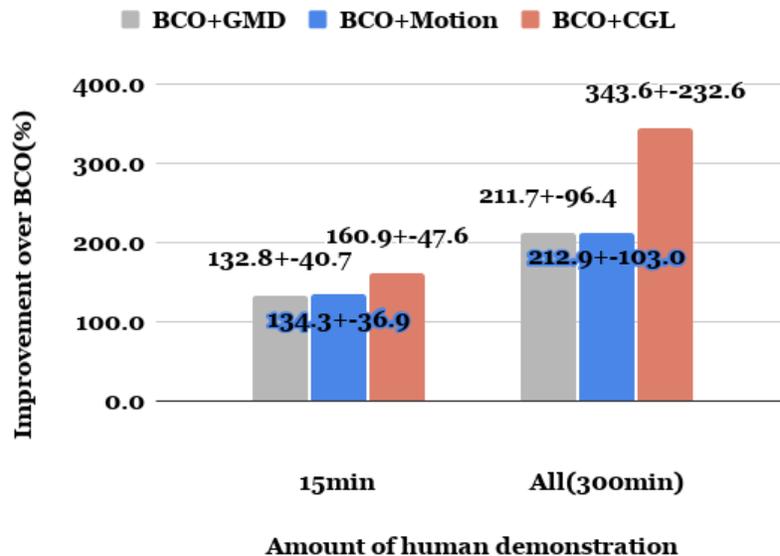


Figure 3.19: Average (across 20 games) improvement over the BCO baseline. Results are presented as mean±standard error of the mean (N=20). Individual game scores of all agents can be found in Appendix A.4.

of the 20 games. For these games, the baseline BCO policy model scores zero, and the utilization of gaze is also unable to overcome the shortcomings of the inverse dynamics model. Additionally, we test GMD and CGL with BCO and find that on average across 20 games, CGL outperforms GMD using 15 or 300 minutes of demonstration data (Fig. 3.19).

Prior work has established that human gaze encodes attention which is different from salient regions in a scene (such as motion) [173, 116, 259, 302]. We test whether our proposed approach extracts the additional information from human gaze data, in comparison to what might already be encoded in the visual game state, such as motion. We replace the gaze heat maps used

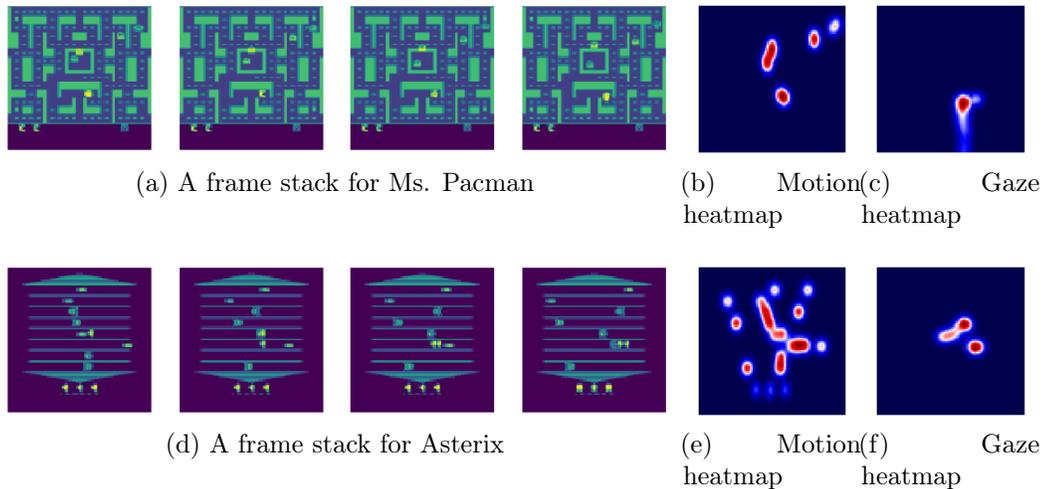


Figure 3.20: Motion in the visual game state, i.e. the difference between the last and first frame in an input image stack, cannot alone explain human attention. This is further highlighted with minimal performance gains when CGL utilizes the motion heatmap instead of the human gaze heatmap.

by CGL with heatmaps representing the normalized motion in an input image frame stack (the difference between the last and first frame in a stack shown in Fig. 3.20). We test this motion-based loss for BCO (Fig. 3.19). Overall, the gains from CGL are much higher than those from motion. Our results are in line with prior work [345] which shows that using optical flow between two frames as the attention map provides only moderate performance improvements in Atari Games. Incorporating both gaze and motion information as an auxiliary loss can be investigated as part of future work.

Improvement over T-REX (%)	T-REX+CGL
30min data	$390.4 \pm 203.5$
300min data	$373.6 \pm 206.5$

Table 3.3: Average (across 20 games) improvement over the T-REX baseline. Result is presented as as mean $\pm$ standard error of the mean (N=20). Individual game scores of all agents can be found in Appendix A.4.

### 3.5.3.3 CGL Improves T-REX

For T-REX, along with full 300-minute human data, we evaluate with 30-minutes human data to compare trajectories from two different human trials. T-REX+CGL outperforms basic T-REX on 15 out of 20 games both with 30-minutes and 300-minutes worth of training data. The improvement due to CGL is **390.4%** with 30-minutes of data, and **373.6%** with 300-minutes of data (Table 3.3). There are four games in that T-REX achieves scores of zero but T-REX+CGL can achieve non-zero scores. To the best of our knowledge, CGL is the first method to augment the learning of an IRL algorithm with human gaze.

### 3.5.3.4 Best Performing Models for each Game

We then summarize the best game scores obtained from various algorithms presented above, the results are shown in Table 3.4. We notice that CGL augmented methods achieve the best results in 15 out of 20 games. For comparison, we also show DQN scores [209, 124] as a reference (the evaluation methods are slightly different). With human gaze information (especially with CGL), imitation learning algorithms start to match and even outperform DQN.

Note that DQN is trained with 200M samples per game, while IL methods are at most trained with 360K samples (300-minutes of human data).

### 3.5.3.5 Visualizing CGL Attention

We can analyze whether the CGL agents have successfully learned to pay attention to the critical regions highlighted by human saliency maps in two ways. First, we directly visualize the activation map of the networks trained with and without CGL, which has already been shown in Fig. 3.17 for a trained BCO agent. However, this only shows that the convolutional layer we applied CGL to behaves as expected.

We use a second method to show that the whole trained network has learned to attend to the desired region with CGL. We visualize the attention maps of trained CGL agents with a method commonly used to provide visual interpretations of deep RL agents [106]. The algorithm takes an input image  $I$  and applies a Gaussian filter to a pixel location  $(i, j)$  to blur the image. This manipulation adds spatial uncertainty to the surrounding region and produces a perturbed image  $\Phi(I, i, j)$ . A saliency score for this pixel  $(i, j)$  can be defined as how much the blurred image changes the network output [106]. Doing this for every pixel results in a saliency map that approximates the "attention" of a network. The results for a case of Breakout where CGL outperforms the baseline T-REX method can be found in Fig. 3.21, where the T-REX+CGL agent successfully learned to focus on the ball like the human did, while the T-REX agent did not.

Game	Algorithm (#demo)	Score	DQN Score
alien	AGIL (300min)	<b>2104.7</b>	1620.0
asterix	T-REX+CGL (30min)	<b>66445.0</b>	4359.0
bank_heist	BC-2ch (300min)	174.3	<b>455.0</b>
berzerk	BCO+CGL (15min)	<b>687.67</b>	585.6
breakout	T-REX+CGL (300min)	<b>438.4</b>	385.5
centipede	T-REX+CGL (30min)	<b>20762.5</b>	4657.7
demon_attack	T-REX+CGL (300min)	<b>17589.0</b>	12149.4
enduro	BC+CGL (300min)	445.1	<b>729.0</b>
freeway	BC-2ch (300min)	<b>31.4</b>	30.8
frostbite	BC+CGL (30min)	<b>5897.7</b>	797.4
hero	BC+CGL (15min)	19023.2	<b>20437.8</b>
montezuma	BC+CGL (300min)	<b>1720.0</b>	0.0
ms_pacman	BC+CGL (30min)	2739.7	<b>3085.6</b>
name_this_game	AGIL (300min)	5817.0	<b>8207.8</b>
phoenix	AGIL (300min)	5140.0	<b>8485.2</b>
riverraid	T-REX+CGL (300min)	7370.0	<b>8316.0</b>
road_runner	BC+CGL (300min)	33510.0	<b>39544.0</b>
seaquest	T-REX+CGL (30min)	759.3	<b>5860.6</b>
space_invaders	T-REX+CGL (300min)	1563.7	<b>1692.3</b>
venture	BC+CGL (15min)	<b>376.7</b>	163.0

Table 3.4: A summary of the best game scores obtained using different imitation learning algorithms. DQN scores are from no-op starts evaluation regime table of [123] , except for game Riverraid [209]. With human gaze information (especially with CGL), imitation learning algorithms start to match and even outperform DQN.

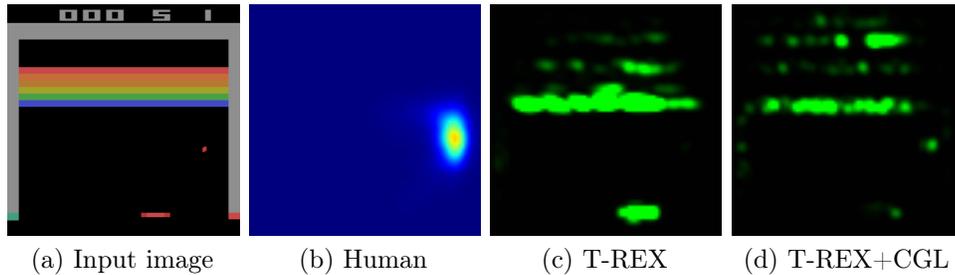


Figure 3.21: CGL guides a convolutional network to focus on parts of the state space (c,d) which the human attends to (b) for Breakout (a). The attention maps of deep networks are generated using the method proposed by [106].

### 3.5.4 Reducing Causal Confusion with Human Attention

Discriminative models for IL such as BC are non-causal, i.e. the training procedure is unaware of the causal structure of the interaction between the demonstrator and the environment. Causal misidentification is the phenomenon where cloned policies fail by misidentifying the causes of the demonstrator’s actions. A very problematic effect of the distributional shift in BC can lead to causal misidentification. This is exacerbated by the causal structure of sequential action: the very fact that current actions cause future observations often introduces complex new nuisance correlates.

Prior work on understanding causal confusion in IL [73] uses past action information (often correlated with current action) to identify if the IL algorithm is in a causal confusion trap. To understand the performance gains of CGL, we investigate if it disambiguates the intent of the user in the demonstrated actions by eliminating causal confusion. We overlay the four-frame image stack (state) with actions from the last frame in the previous stack (Fig. 3.22). This

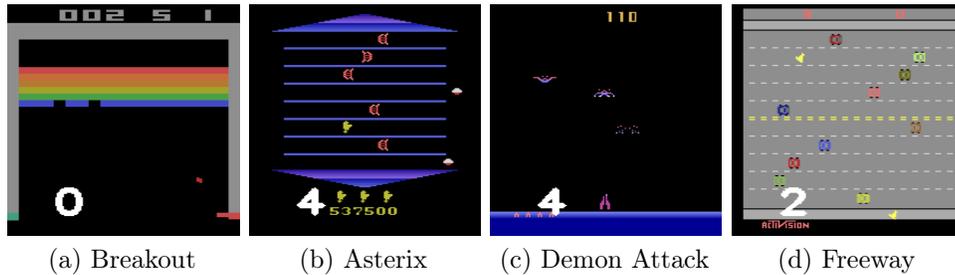


Figure 3.22: Confounded states with past actions to test reduction of causal confusion with CGL. The design follows [73].

lays a causal confusion trap for the IL agent. If the agent can ignore the new correlated action information that is part of the state space, it hints towards the fact that the agent learns to ignore those features and perform better empirically. We find that on average (across 20 games), CGL agents suffer less when trained with confounded data compared to the BC baseline (**-34.0%** versus **-47.8%**), and still perform much better (by 571%) than BC agents. Detailed results are available in Appendix A.4. This hints to the fact that in addition to directing the attention of the network to learn a better mapping between states and actions, part of the gains from using human gaze data with CGL can come from reducing causal confusion.

### 3.5.5 Summary

To resolve the four issues with AGIL-like methods, we introduced an auxiliary coverage-based gaze loss (CGL) term which guides the training of any imitation learning network with convolutional layers. Our experiments showed improved performance on several Atari games over standard imitation

learning algorithms. Our approach provides these gains without requiring gaze prediction at test time or increasing the model complexity of existing algorithms. This work confirms prior research showing gaze can help extract more information from a demonstrator than traditional state-action pairs, bridging the gap in performance between IL and RL agents.

## 3.6 Human versus Machine Attention<sup>9</sup>

### 3.6.1 Background and Motivation

The Atari-HEAD dataset provides us a unique opportunity to understand deep reinforcement learning (RL) models, which are RL agents that use deep neural networks (DNNs) as function approximators.

Researchers have devoted much effort to understanding DNNs. Since DNNs are partially inspired by the biological nervous systems, researchers have compared these neural networks with human brain and sensory systems by asking two typical types of questions. The first one considers representation: How similar are the visual features learned by DNNs and humans when performing the same tasks? The second one concerns explainability: How do similarities and differences in these learned features explain DNNs' performance on their tasks?

---

<sup>9</sup>This section of work is based on the following publication: Ruohan Zhang, Bo Liu, Yifeng Zhu, Sihang Guo, Mary Hayhoe, Dana Ballard, and Peter Stone. Human versus machine attention in deep reinforcement learning tasks. arXiv preprint arXiv:2010.15942, 2020. The dissertator is the first author, and takes the leading role in conceiving and designing the analysis, collecting the data, contributing analysis tools, performing the analysis, and writing the paper.

The first question motivates a seminal line of research that compares features learned by DNNs with features learned by brains. In computer vision, a linear relationship has been found between features learned by convolutional neural network layers and neural responses of visual cortex [83, 339, 340]. In language learning, a similar connection has been found between deep language models and cortical areas [141, 149] (see Section 3.8.1.4 for a review). However, this type of comparison has just emerged in decision-making research [205, 67], which motivates us to compare DNNs trained for deep reinforcement learning (RL) tasks with human decision-making. In particular, we ask: For a given task, do deep RL agents and humans agree on what visual features are important? In other words, do agents pay *attention* to the same visual features as humans do?

The explainability question is motivated by the Explainable AI (XAI) paradigm for deep RL agents [126, 6, 241] (see Section 3.8.1.4 for a review). Deep RL agents still make mistakes, which researchers are interested in providing explanations for. However, these agents often learn a mapping from raw images to an action end-to-end where it is not clear why a particular decision is made. Our work addresses the explainability of these black-box models: Do RL agents make mistakes because they fail to attend to important visual features that matter for making the correct decision? An expert human’s attention data could serve as a useful reference for identifying important visual features, which has been validated in computer vision research for object recognition tasks [216].

To answer these two questions, we situate our research in the domain of Atari games [31]. Two recent lines of research have made our work possible. On the one hand, tools have been introduced to generate visual interpretations of RL agents [107]. Through these tools, one can visualize an RL agent’s *saliency map* given an image, which is a topographically arranged map that assigns an importance weight to each image pixel and is often treated as the “attention map” of the RL agent. On the other hand, Atari-HEAD provides a human attention dataset [363], which makes a comparison study feasible.

In this section, we present the first study that compares human attention with the RL agent’s attention [356]. We analyze how learning and hyperparameters of the RL algorithm affect the learned features and saliency maps. We show that RL agents make mistakes partly because they fail to attend to the correct visual targets. We further study whether RL agents’ attention generalizes to states that these agents have never visited. We conclude by discussing insights gained for RL researchers in both cognitive science and AI.

### 3.6.2 Method

We now discuss our approaches for obtaining the human attention model, training RL agents, and extracting attention information from the trained RL agents. We then discuss how we select data for comparing human versus RL attention, and define the comparison metrics.

### 3.6.2.1 Human Attention Data and Model

We use human expert gaze data from the Atari-HEAD dataset [363]. The original game runs continually at 60Hz [31], a speed that is challenging even for professional gamers. Human eye movements were rushed and inaccurate at this speed hence could not serve as a useful reference. Recall that in Atari-HEAD, however, the human data were collected in a semi-frame-by-frame mode, a design that allowed enough time for the players to attend to all relevant objects on the screen and make decisions.

To get human saliency maps for the data generated by RL agents, we need an accurate human attention model. We did not use human states due to a state distribution mismatch. RL agents make many more mistakes than humans and cannot reach the late stages of the games like human experts. This creates a state distribution that does not match human data, which contains mostly good states. Having a human attention model allows us to perform comparisons in states encountered by RL agents only which is especially important when we later analyze RL agents' failure states. However, if the focus is on using RL to model human decisions, one can take our approach with human states [67].

As a control analysis, we include two attention models in addition to human gaze. The first one captures the motion information, measured by Farneback optical flow between two consecutive images [87]. The second model captures salient low-level image features, including color, orientation, and intensity (weighted equally), computed by the classic Itti-Koch saliency model [144].

### 3.6.2.2 Reinforcement Learning Agent and Attention Model

As a case study, we use a popular deep RL algorithm named Proximal Policy Optimization (PPO) [269] with default hyperparameters [127]. For each experimental condition (discussed later), we train 5 models with different random seeds. For the Atari gaming environment, we use the basic version that has no frame skipping and no stochasticity in action execution (NoFrameskip-v4 version). We select six popular Atari games.

We use a perturbation-based method [107] to extract attention maps from PPO agents, which has been validated by several subsequent studies [113, 276]. The algorithm takes an input image  $I$  and applies a Gaussian filter to a pixel location  $(i, j)$  to blur the image. This manipulation adds spatial uncertainty to the region around and produces a perturbed image  $\Phi(I, i, j)$ . A saliency score for this pixel  $(i, j)$  can be defined as how much the blurred image changes the policy  $\pi$ :

$$S_{\pi}(i, j) = \frac{1}{2} \|\pi(I) - \pi(\phi(I, i, j))\|^2 \quad (3.8)$$

Instead of calculating the score for every pixel, [107] found that computing a saliency score for pixel  $i \bmod 5$  and  $j \bmod 5$  produced good saliency maps at lower computational cost for Atari games. The final saliency map  $P$  is normalized as  $P(i, j) = \frac{S_{\pi}(i, j)}{\sum_{i, j} S_{\pi}(i, j)}$ .

### 3.6.2.3 Comparison Metrics

Next, we compile a set of game images to compute human and RL saliency maps. For each game, we let a trained PPO agent (with default hyperparameters) play the game until terminated, and uniformly sampled 100 images from the recorded trajectory. We will refer to this set of images as the standard image set.

We then define two metrics for comparing saliency maps: Pearson’s Correlation Coefficient (CC) and Kullback-Leibler Divergence (KL) (Section 3.3). Again, let  $Q$  denote the human saliency map predicted by the human attention network. CC is between  $-1$  and  $1$  captures the linear relation between two distributions  $Q$  and  $P$  (Eq. 3.3). CC penalizes false positives and false negatives equally.

However, we may not want to penalize the RL agent if it attends to regions that the human gaze model is not currently focused on. The human gaze data only reveals their “overt” attention, and humans can still pay “covert” attention to entities in the working memory [239, 268], as we have discussed in Section 3.5. Thus we need a second metric that penalizes the agent only if it *fails* to pay attention to human attended regions, or equivalently, a metric that is sensitive to false negatives if we treat human attention as the ground truth. KL divergence (Eq. 3.2) is an ideal candidate in this case [53]. The  $\epsilon$  is a small regularization constant (chosen to be  $2.2204e-16$  [53]) and determines how much zero-valued predictions are penalized.

### 3.6.3 Results

To make meaningful comparisons, we first ensure that the human attention model is accurate, and PPO agents’ attention maps are consistent over repeated runs. We then compare human attention with PPO attention that is obtained from different learning stages and from agents that are trained with different discount factors. We then analyze PPO agents’ attention in failure and unseen states. At last, we show comparison results for other deep RL algorithms.

**Accuracy of human attention model** We implement the convolution-deconvolution gaze prediction network (Fig. 3.5) [363] to generate a human saliency map for image  $I_t$  at timestep  $t$ , given a stack of four consecutive images  $I_{t-3}, I_{t-2}, I_{t-1}, I_t$  as input. We use 80% gaze data for training and 20% for testing. The network model can accurately predict the human gaze. On testing data, we obtained Area under ROC Curve (AUC) score of  $0.968 \pm 0.005$ , CC of  $0.562 \pm 0.030$ , and KL of  $1.411 \pm 0.114$  ( $n = 6$ ), averaged over all games.

**Consistency of RL attention** We then show that saliency maps of the RL agents trained under the same experiment setting, provided by [127], are highly consistent, despite the stochasticity in the training process. The stochasticity is controlled by a random seed which is used to initialize both the game environment and the network. For each game, we use 5 random seeds (0-4), train an agent using each seed, and generate 5 saliency maps with these trained

agents. For each image in the standard set, we compute pair-wise CCs between these 5 saliency maps (10 CCs in total). The average value for these 10 CCs, across 100 images and 6 games, is  $0.924(\pm 0.001, n = 6000)$ . Given such high consistency, the saliency maps we use later are the averaged results of these 5 saliency maps.

### 3.6.3.1 The Effects of Learning

So far we have verified that the gaze model can accurately predict human attention and that RL attention is consistent across repeated runs. We now address the primary research question: How similar are the visual features learned by RL agents and humans when performing the same task? We first study how the attention of PPO agents evolves over time, compared with human attention. For each game, we saved neural network weights at different time steps during training. Then we use these saved models to generate saliency maps on the standard image set.

Fig. 3.23 shows the aggregated results for all games. CC values and (negative) KL values between humans and RL increase during learning, indicating that the RL agents' attention gradually becomes more human-like. We visualize the change of RL attention and human attention in Fig. 3.24. Networks without any training (time step 0) have saliency maps that are positively correlated with humans (CC= 0.170). For example, the second column in Fig. 3.24 shows that these networks are already sensitive to low-level salient visual features without any learning, which is consistent with previous findings [340, 107].

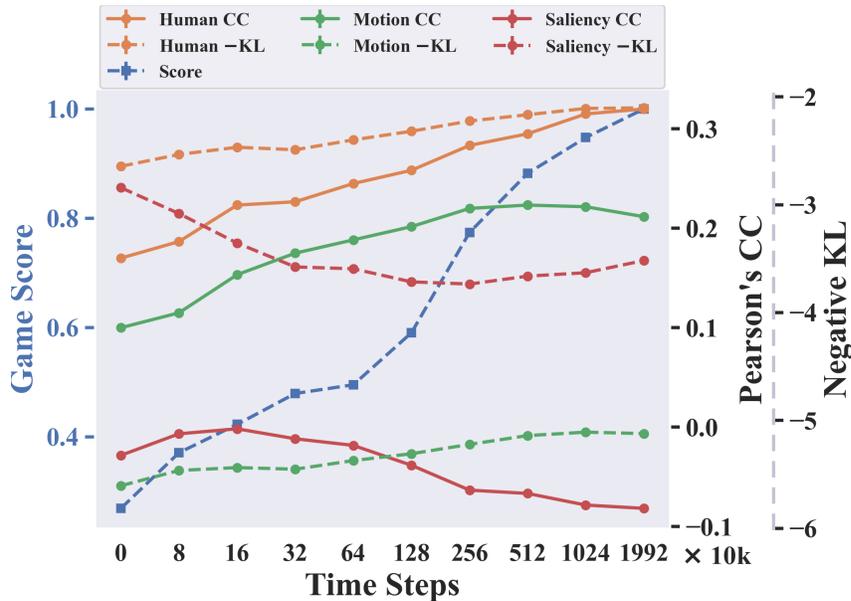
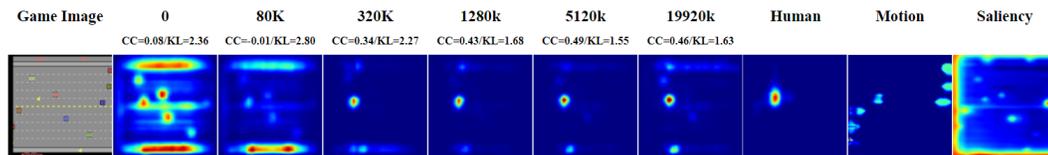


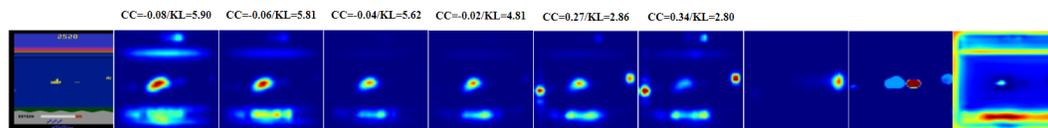
Figure 3.23: Changes in human and RL attention similarity across learning time steps. PPO agents gradually learn to pay attention to important visual features and develop become more human-like. We also show two control attention models: motion (optical flow) and saliency (low-level image features). The x-axis is log scale and KL values are negated for better visualization. The CC, KL, and score results for individual games, as well as more examples, can be found in Appendix A.5.1.

After training, human and RL saliency maps are more positively correlated (CC= 0.320), meanwhile RL and human are more similar to each other than the control models (higher CC/-KL values after training).

We also show aggregated game performance in normalized game scores in Fig. 3.23. For each game, we normalize the game scores obtained during learning (averaged over 50 episodes) by dividing them by the final scores. We find a strong positive correlation between the human CC/KL values and the game score (average Pearson’s correlation coefficient of 0.813/0.790). The



(a) Freeway: The RL agent gradually learns to focus its attention on the yellow chicken crossing the highway.



(b) Seaquest: The agent learns to attend to the yellow enemy on the right which poses a threat to the yellow submarine in the middle.

Figure 3.24: Attention of RL agents changes during learning and becomes more human-like. CC/KL values are calculated between the RL agent’s attention map and the human attention map.

correlations for all games are statistically significant ( $p < 0.05$ , Appendix A.5.1), indicating that small changes in similarity with human attention are reliable predictors of performance change. For comparison, the correlation values between CC/KL and game score for motion baseline are 0.404/0.251, for saliency baseline are  $-0.258/ -0.688$ .

This result sheds light on an important attention research topic: bottom-up versus top-down attention. The two sides debate how much human or machine attention is driven by bottom-up image features captured by the saliency model [144], and how much it is driven by top-down task signals such as reward [259, 40]. Our result suggests that learning is a key factor in this debate. In the early stages, attention is more driven by image features, indicated by the higher similarity between RL attention and saliency baseline at the

beginning. Then top-down reward signals shape the attention during learning by making reward-associated objects more salient and irrelevant objects less salient, as shown in Fig. 3.24.

### 3.6.3.2 The Effects of Discount Factors

We then analyze how hyperparameters of the PPO algorithm affect the attention of the trained agents. We have seen that reward shapes attention during training, therefore a reasonable hypothesis is that varying reward-related parameters will likely affect attention. The parameter we can vary in these games is the discount factor  $\gamma \in [0, 1)$ , which determines how much the RL agent weighs future reward compared to immediate reward. The default  $\gamma$  is 0.99 for all games [127]. We train PPO agents with  $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9, 0.9999\}$  and generate saliency maps on the standard image set. The results are shown in Fig. 3.25. Overall, the RL attention is most similar to humans when  $\gamma = 0.7$  or 0.9. From the visualizations in Fig. 3.26 we can see that with small  $\gamma$ s, the agents tend to be myopic and only focus on very few objects. With high  $\gamma$ s, the agents divide their attention into several objects. Since the human gaze model only captures the overt attention and involves typically 1-3 objects, intermediate  $\gamma$  values produce saliency maps that are similar to humans.

For each  $\gamma$  value, we normalize the game score by dividing it by the score obtained by the  $\gamma = 0.99$  agent. For most games, this default value has the best performance. However, the performance increases with CCs and

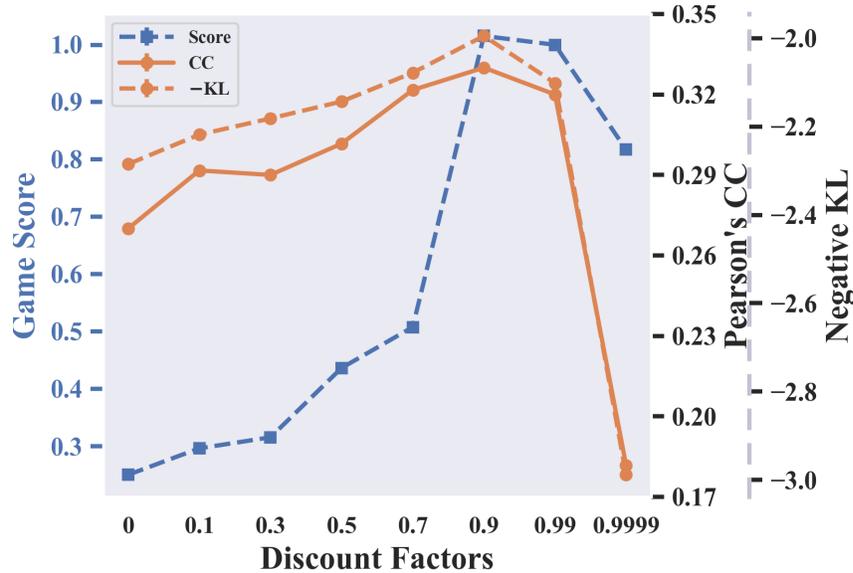
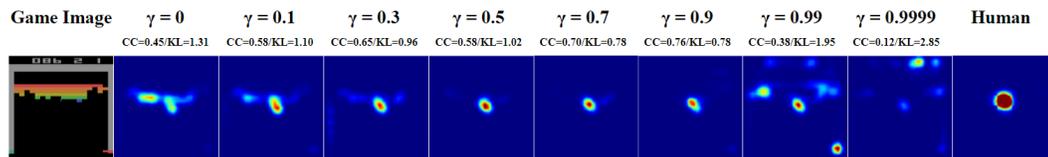


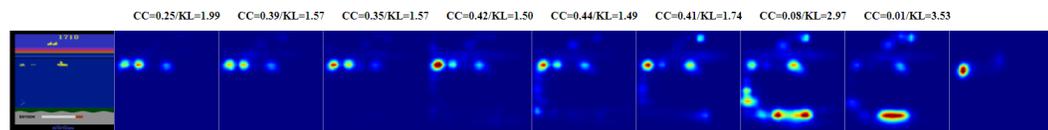
Figure 3.25: Changes in human and RL attention similarity across different discount factors. Choosing different discount factors affects the RL agent’s attention and performance. The CC, KL, and score results for individual games, as well as more examples, can be found in Appendix A.5.2.

negative KL values only up to  $\gamma = 0.7$  in Fig. 3.25. Beyond that, the RL agents need to learn to attend to objects that matter in the long-term, which are likely to be captured by human covert attention and do not show up in our human attention model.

Nonetheless, deviating from the default  $\gamma = 0.99$  can lead to better performance. In Seaquest, the agent with  $\gamma = 0.9$  achieves a 15% higher score than the default agent. Fig. 3.26b shows that with a lower  $\gamma = 0.9$ , the agent can focus, like humans, on an immediate threat from the left. Setting  $\gamma = 0.99$  or 0.9999 distracts the agent to attend to the oxygen bar at the bottom that is important in the long run but less urgent now. A similar result was found



(a) Breakout: The human pays attention to the ball.



(b) Seaquest: The human pays attention to an approaching enemy from the left and oxygen level at the bottom.

Figure 3.26: Effect of different discount factors on Ms.Pac-Man and Seaquest agents’ attention. Agents trained with intermediate discount factor values have saliency maps that are more human-like.

for Ms.Pac-Man with an 18% higher score from the  $\gamma = 0.9$  agent than the  $\gamma = 0.99$  agent.

This result has an important implication. Atari games are episodic tasks with true  $\gamma = 1$  but lower  $\gamma$  values often lead to better performance in practice [269, 209]<sup>10</sup>. Fig. 3.25 shows that  $\gamma = 0.9999$  agents perform poorly. We have verified that all  $\gamma = 0.9999$  agents (5 random seeds) converged after 200M samples, although to sub-optimal policies. Again, the reason is that being a little myopic helps the agent focus on the most urgent targets. This result provides another reason, from a perception perspective, for why RL agents need to adjust their planning horizon by reducing the discount factor—confirming

<sup>10</sup>Strictly speaking,  $\gamma$  is a property of the MDP, not of the agent. Performance across MDPs with different  $\gamma$ s are not directly comparable in this sense. Varying discount factor and clipping reward are common reward engineering designs that alter the true MDP return to achieve better performance in practice.

the theoretical results provided by [154].

### 3.6.3.3 Failure States Analysis

We now turn to the second research question that concerns explainability in deep RL: Why do deep RL agents make mistakes? Did RL agents fail to attend to the right objects, or did they attend to the right objects but make wrong decisions? We compile a second image set, named failure image set, by recording the game frames right before the RL agent loses a “life” which incurs a large penalty. We locate 100 such instances for each game. Freeway is excluded since the PPO agent learned a nearly optimal policy.

Fig. 3.27 shows two failure cases for Breakout. In the first case, the attention map of the RL agent is extremely different from the human’s, and in the other, they are highly similar. In the first case, the RL agent failed to attend to the ball as the human model did, hence it failed to move the paddle to catch the ball in the next frame (it is still possible to catch that ball). In the second case, both the agent and the human attend to the ball. In this way, we can interpret failure cases using human attention as a reference. If RL attention and human attention are more different in failure than normal states, there is likely to be a perception problem.

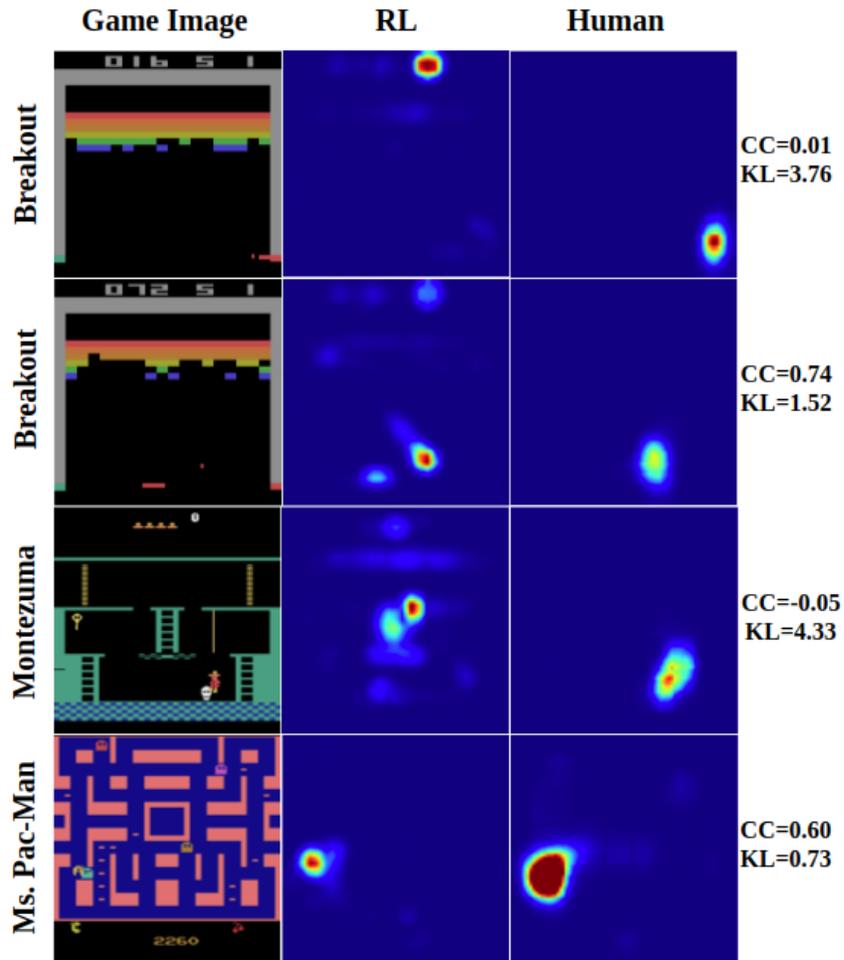


Figure 3.27: RL vs. human attention in states where RL agents made mistakes. We show examples of RL and human saliency maps for Breakout (top: wrong attention; bottom: right attention but wrong decision), Montezuma’s Revenge (wrong attention), and Ms.Pac-Man (right attention but wrong decision).

Fig. 3.28 shows the quantitative results for all games. For Montezuma’s Revenge, perception is a problem. The RL attention in failure states becomes less similar to human attention compared to the standard image set, indicated by significantly decreased CC/KL values. Fig. 3.27 shows that the agent fails to attend to the enemy as the human does.

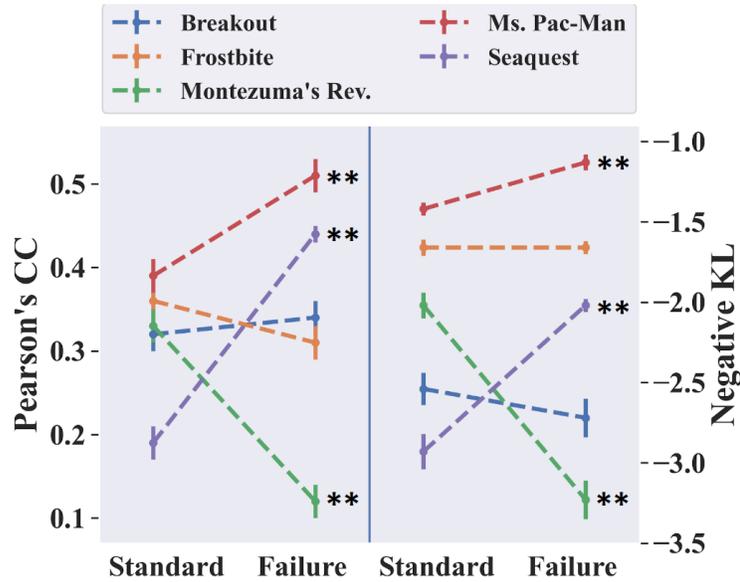


Figure 3.28: How attention similarities change in failure states compared to normal states. \* indicates a significance test result of  $p \leq 0.05$ ; \*\* indicates  $p \leq 0.01$ . Error bars are the standard errors of the mean ( $n = 100$ ).

On the contrary, for Ms. Pac-Man and Seaquest, perception is *not* a major problem. Fig. 3.28 shows that, in failure states, attention maps of RL agent and human are more similar than in the normal states, suggesting that they generally agree on the objects to be attended to. These are often situations with fewer objects that are more dangerous (see Appendix A.5.3). Hence it is easy for RL agents to attend to the right object but the decision

is hard. For example, in Fig. 3.27 Ms.Pac-Man, the agent attended to the Pac-Man and the enemy ghost similarly to humans. However, the RL agents made the wrong decision and ran into the ghost. For Breakout and Frostbite, the overall differences between failure and normal states are not significant, but one can perform a case-by-case analysis as in Fig. 3.27.

We conclude that the mistakes made by the RL agents are sometimes related to perceptual errors. Similarity measurements with human attention can help identify and interpret the failures made by RL agents.

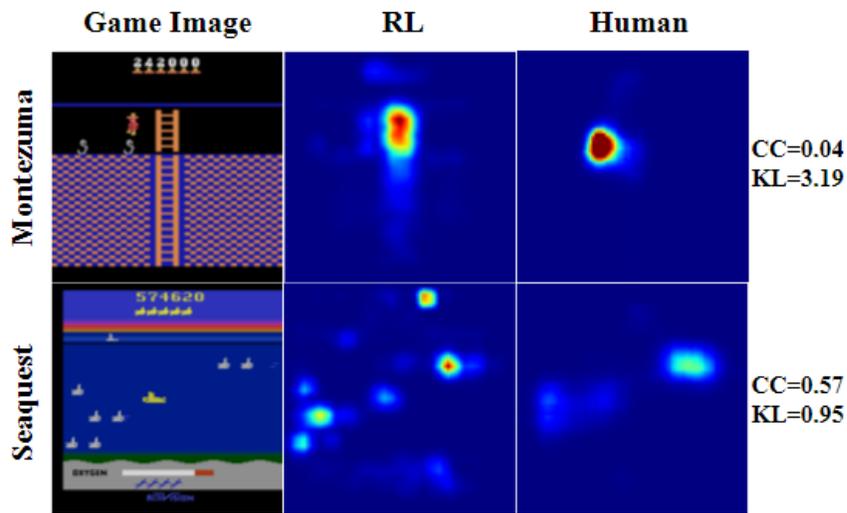
#### 3.6.3.4 Unseen Data Analysis

Next, we study whether the RL agent’s attention generalizes to unseen states. The unseen states are late-game states obtained from human experts’ data [363] which RL agents have not encountered. We refer to this set as the unseen image set (100 images per game). Again, Freeway is excluded since the agent is nearly optimal.

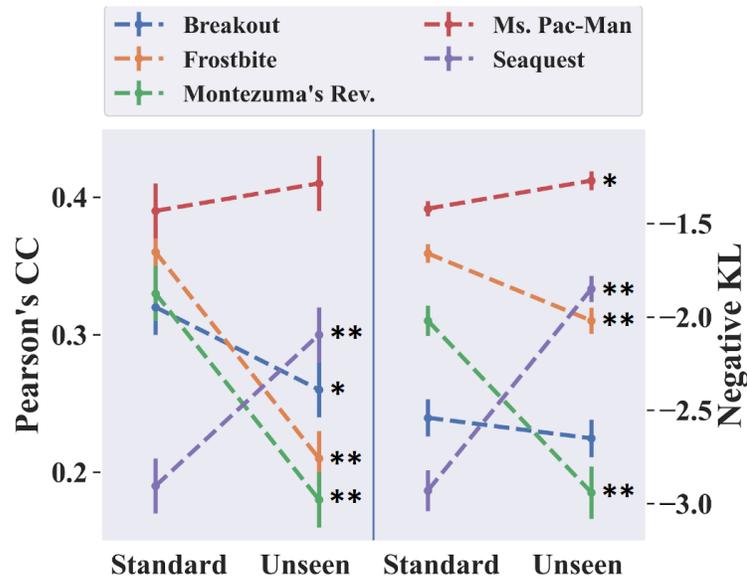
Fig. 3.29b shows the results. For Frostbite and Montezuma’s Revenge, the similarities drop significantly, mostly due to new objects that the agents have never encountered. Fig. 3.29a shows an example for Montezuma’s Revenge. The agents attended to the ladder, a previously seen object, but failed to attend to a new enemy object on the left like the human did. For Seaquest and Ms. Pac-Man, the RL attention and human attention are more similar on unseen data. This is because there are no new objects in these unseen states – only objects move faster and appear in larger numbers. The player often encounters

dangerous states that are close to failure, similar to the ones in Section 3.6.3.3. Therefore we observe a similar increase in similarity for Ms.Pac-Man and Seaquest. Breakout is an interesting case. The CC value drops significantly due to unseen spatial layouts of the objects. The KL does not change much because there are no new objects so the agent maintains its attention on the ball and the paddle. More examples are in Appendix A.5.4.

The results suggest that the obstacle to achieving human-level performance for certain games is first a perception challenge – the agents need to learn to recognize, attend to, and then learn to act upon new objects. This is easy for humans due to their prior knowledge but challenging for RL agents [172, 310, 82]. For the other games, it is a decision-learning problem – the agent’s attention is fairly generalizable and it needs to learn a good policy for challenging states.



(a)



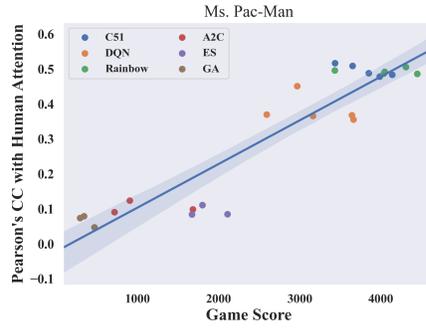
(b)

Figure 3.29: Human versus RL attention in states that RL agents have not seen. (a) Unseen states. for Montezuma’s Revenge (top), the RL agent’s attention is very different from human attention due to a new object on the left. For Seaquest (bottom), they are similar. (b) How attention similarities change in unseen states compared to seen states in different games.

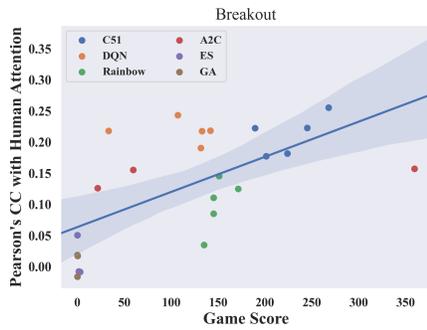
### 3.6.3.5 Other Atari Agents

The above analyses were done for a particular RL algorithm—PPO. Next we apply our method to other RL algorithms, including C51 [29], Rainbow [124], DQN [209], and A2C [207], as well as two evolutionary algorithms, GA [294] and ES [264]. We use trained models from Dopamine [57] and Atari Model Zoo [295] in which each algorithm has 3-5 trained models. Note algorithms such as DQN outputs state-action values instead of policy hence they will attend to game scores at the top or the bottom of the screen. To ensure a fair comparison for policy-based and value-based algorithms, we ignore attention that is on game scores when doing comparisons (by cropping out that part of the image when calculating the similarity), following the standard approach [107, 113, 138].

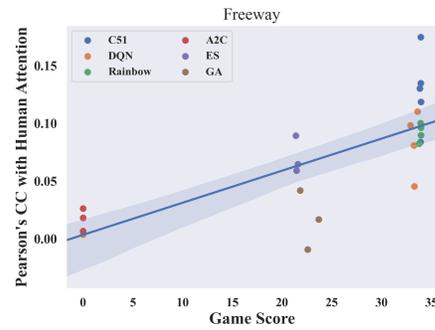
Fig. 3.30a shows the result for Ms.Pac-Man. There is a strong positive correlation ( $r(22) = 0.927, p < 0.001$ ) between model performance (in terms of the game score, averaged over 50 episodes each) and similarity measurement (in terms of CC with human attention on the standard image set). The average correlation coefficient for five games is 0.631 (excluding Montezuma since most algorithms have zero scores). Fig. 3.30 shows the result for Breakout, Freeway, Frostbite, and Seaquest. This result suggests that the positive correlation between model performance and similarity with human attention generalizes to other deep RL algorithms, hence we expect the rest of the analyses we performed with PPO, such as failure diagnosis, can be done for these algorithms as well.



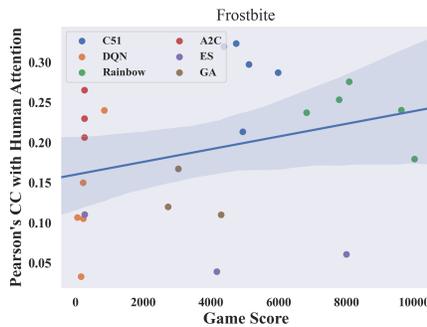
(a)



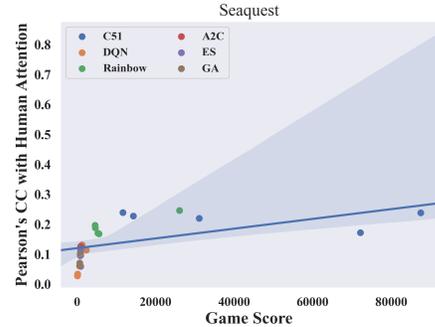
(b)



(c)



(d)



(e)

Figure 3.30: The relation between the similarity with human attention and algorithm's performance. The line shows the linear regression line fitted to the data point and the shaded area is the 95% confidence interval. The correlation coefficients between the similarity measurement and game score are:  $r(22) = 0.634, p < 0.001$  for Breakout,  $r(22) = 0.743, p < 0.001$  for Freeway,  $r(22) = 0.298, p = 0.158$  for Frostbite, and  $r(22) = 0.553, p < 0.01$  for Seaquest.

### 3.6.4 Discussion

We provide visual explanations for deep RL agents using human experts' attention as a reference. We have discussed how RL attention develops and becomes more human-like during training, and how varying the discount factor affects learned attention. We show that human attention can be useful in diagnosing an RL agent's failures. We have also identified further challenges in closing the performance gap between human experts and RL agents. Our analysis is restricted to saliency map comparisons, but other approaches are possible for measuring the similarity of representations learned by different RL agents [328]. Our human attention models, all compiled datasets, and tools for comparing RL attention with human attention are made available for future research in this direction.

For researchers who are interested in RL algorithms, we have gained two important insights. Firstly, our results provide visual explanations for the agents' performance when varying the discount factors and highlight the importance of choosing proper planning horizons with appropriate discount factors. Recent works confirm this by showing that it is beneficial to have an adaptive discount factor [15] or multiple discount factors [88]. Secondly, failure analysis could identify states where agent's attention drastically differs from expert humans. These are the states that may need human intervention/correction in a human-in-the-loop RL paradigm. By publishing our human gaze prediction model researchers can diagnose their algorithms with the states of their interest.

For researchers who are interested in using RL as models for cognition, it

is perhaps both surprising and encouraging to see that RL agents trained from scratch with only images and reward signals can develop attention maps that are similar to humans, especially when considering that they have very little prior knowledge. This result is similar to previous research that shows CNNs trained from image classification tasks can learn features that are similar to the ones in the visual cortex [83, 339, 340]. Consistent with previous findings, we show that model task performance and feature similarity are highly correlated [340]. Our results are complementary to the recent findings using human fMRI data when playing Atari games [67], suggesting that deep RLs can learn biologically plausible representations and can be used as models for human gaze, decision, and brain activities.

### **3.7 Human Attention-Guided Reinforcement Learning**

In Section 3.6 we have compared human attention and RL agent’s attention. Since the task performance and similarity to human attention are highly correlated, one could use human attention as prior knowledge to guide the learning process of RL agents, e.g., by encouraging them to attend to the correct objects early in the learning process. This could be especially helpful for games like Seaquest, in which the agent has not learned to focus on the right object after 5120k time steps (Fig. 3.24). Two human studies using Atari games suggested that prior knowledge, such as perceptual prior, is why humans learn faster and better in these games [310, 82]. Then the next question is how to incorporate human attention into DNNs which has been studied in

several computer vision tasks [243], and is successful in training deep imitation learning agents (Section 3.4, 3.5) [361, 354, 268].

Given these results, a natural next step is to incorporate visual attention into deep RL. The assumption made here is that visual features that capture human attention are likely to be informative for deep RL agents. Deep Q-network has demonstrated the effectiveness of end-to-end learning of visuomotor tasks [209]. However, for games such as Seaquest and MsPacman—which typically involve multiple tasks—the performance is below the human level. Besides, DQN takes millions of samples to train. The above issues could be potentially alleviated by combining human attention and deep RL where the attention model can help extract features to speed up learning and to indicate task priority.

### 3.7.1 Attention-Guided Reinforcement Learning<sup>11</sup>

Here we will try to build a hybrid imitation learning–reinforcement learning agent. The agent will learn attention from human gaze and subsequently use the learned model to help itself learn to perform the task via deep reinforcement learning. However, the attention-guided RL has not shown improvement over standard RL in Atari games so far, in terms of both learning speed and end performance [345]. One possibility is that we have not found the

---

<sup>11</sup>This section of work is based on the following publication: Liu Yuezhong, Ruohan Zhang, and Dana H Ballard. An initial attempt of combining visual selective attention with deep reinforcement learning. arXiv preprint arXiv:1811.04407, 2018. The dissertator is the second author, and contributes in conceiving and designing the analysis, collecting the data, contributing analysis tools, and writing the paper.

best neural network architecture or hyperparameters for the combined model. Another possibility is that, in these Atari games, perception learning may not be the most challenging component, i.e., the RL agent could learn to perceive relatively fast, but decision learning may take significantly more time. In other words, attention helps the most when perception is difficult.

To test this hypothesis, we use a simplified environment [345] called Catch. Figure. 3.31a shows the original version of the game. A ball falls from the top of the screen, and the agent controls the paddle at the bottom to catch the ball. To be more specific:

- State space:  $20 \times 20$ , with only black and white (0/1) pixels. The ball is of size  $1 \times 1$  and the paddle is  $3 \times 1$ .
- Initial state: at the beginning of each episode, the ball and the paddle are placed at a random position at the top and the bottom row respectively.
- Action space: 3 actions, move the paddle 1 pixel left, stay, or move right.
- Dynamics: the ball falls at the speed of 1 pixel per timestep.
- Reward: the player only receives reward at the end of episodes, +1 if the paddle catches the ball successfully, 0 otherwise.

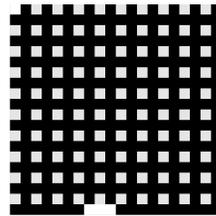
To make perception more difficult, we modify the original Catch environment to include background noise. A latticed background that has a pixel value of 0.9 is added to the original image, as shown in Fig. 3.31b. Here, we use

hand-crafted visual attention information. The attention map marks the true position of the ball and the paddle. We use Deep Q-Network [209] as the RL agent. We use the two-channel AGIL policy network to incorporate attention by replacing behavior cloning loss with the Q-learning loss.

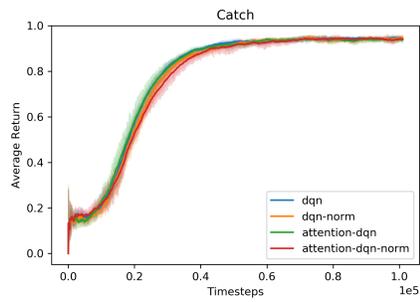
In the original Catch environment, RL with or without attention performs similarly (Fig. 3.31c). All methods get more than 0.9 averaged return (which also denotes an over 90% successful rate; we average past 100 episodes and set the final exploration rate  $\epsilon = 0.1$ ) after 40000 timesteps, indicating the agents have learned the task successfully. However, for the Noisy environment, only the method with visual attention can still achieve a similar performance - over 90% success rate (with batch normalization to stabilize the learning), as shown in Fig. 3.31c. Other methods can only succeed in about 50% of the trials - which is exactly the probability that the ball is initialized in the columns without noisy background. This simple experiment illustrates that attention information helps the most in the tasks in which perception itself is challenging. For our experiments on Atari games, this may explain why attention helps RL on some games but not the others.



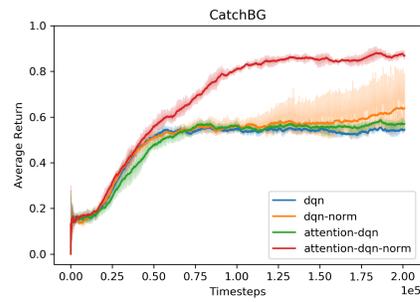
(a) Catch



(b) Noisy Catch



(c) Learning curve for Catch



(d) Learning curve for Noisy Catch

Figure 3.31: The effect of attention on learning to play a simple game called "Catch". Attention helps the most when significant amount of noise is presented and perception itself is a difficult problem.

### 3.7.2 Guiding Reinforcement Learning with Evaluative Feedback and Attention Information<sup>12</sup>

There are at least three important lessons we have learned so far. (1) Agents without working memory need more information than those revealed by the human overt attention. (2) Attention is a useful source of information as an inductive bias that can be used as a regularizer in learning. Its benefit is most evident when data is scarce and sample efficiency matters. (3) Attention is most useful when perception is challenging.

These lessons have motivated the EXPAND (Explanation Augmented Feedback) framework, in which we incorporate attention-like information into human-in-the-loop RL [110]. We focus on the task of learning from evaluative feedback, in which the human trainer provides binary evaluative “good” or “bad” feedback for queried state-action pairs when watching an RL agent learning to perform the task (Section 2.3.2.1). Since human feedback is expensive, sample efficiency is critical. Attention information can naturally be used to extend the communication channel between humans and agents in this interactive setting. Instead of using human gaze, we ask humans to annotate all important visual features by drawing bounding boxes around them. Hence the attention information includes all relevant features, which can be viewed as a form of visual explanation. We choose three visually challenging tasks with high-

---

<sup>12</sup>This section of work is based on the following publication: Lin Guan, Mudit Verma, Sihang Guo, Ruohan Zhang, and Subbarao Kambhampati. Explanation augmented feedback in human-in-the-loop reinforcement learning. arXiv preprint arXiv:2006.14804, 2020. The dissertator contributes in conceiving and designing the analysis, contributing analysis tools, and writing the paper.

dimensional state space, namely Pixel-Taxi and two Atari games, to evaluate the performance and sample efficiency of this approach.

[110] also proposes a novel context-aware data augmentation method, which differentiates between relevant and irrelevant regions in human explanation by applying multiple perturbations to irrelevant parts. The basic idea is that, if we perturb (e.g., blurring the image with a Gaussian window) the irrelevant image regions (those not being attended), it should not affect the value function outputs by the deep RL network. Hence, we feed the network with perturbed images (in this way augmenting the data) and design a loss function that enforces the invariance during training. We show that our method significantly outperforms deep RL methods, deep RL methods+human feedback (without attention), deep RL methods+human feedback+attention (without data augmentation, like AGIL), as well as deep RL methods+human feedback+attention+previous data augmentation methods. EXPAND requires fewer interactions with the environment (environment sample efficiency) and over 30% fewer human signals (human feedback sample efficiency) [110].

### **3.8 Discussion, Related Work, and Future Work**

We introduce Atari-HEAD, a large-scale dataset of human demonstration playing Atari video games. We have shown that the data could be useful for modeling human gaze and action, enhancing the performance of IL/RL algorithms, as well as understanding the difference between RL and human attention.

### 3.8.1 Related Work<sup>13</sup>

#### 3.8.1.1 Related Work: Similar Datasets

In imitation learning research, the Atari Grand Challenge dataset pioneered the effort of collecting a large-scale public dataset of Atari games [170]. The human demonstration was collected through online crowdsourcing with players of diverse skill levels. Recently, researchers have spent significant effort in building large-scale datasets of human demonstrations in various tasks, including driving [343], playing Minecraft [114], and manipulating simulated robots [198]. Our dataset joins their effort in providing a standard dataset for the RL and IL research community. Gaze prediction was formalized as a visual saliency prediction problem in computer vision research [144]. The gaze data can be collected with an eye tracker while the human trainer is demonstrating the task. Large-scale datasets have enabled deep learning approaches to make tremendous advances in this area. Examples include MIT saliency benchmark [50], CAT2000 [41], and SALICON [153]. However, the traditional saliency prediction task does not involve tasks nor human decisions. How humans distribute their visual attention for dynamic, reward-seeking visuomotor tasks has received less attention in research on saliency. In recent years, researchers

---

<sup>13</sup>This section of work is based on the following survey papers: Ruohan Zhang, Faraz Torabi, Lin Guan, Dana H Ballard, and Peter Stone. Leveraging human guidance for deep reinforcement learning tasks. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 6339–6346. AAAI Press, 2019. R Zhang, A Saran, B Liu, Y Zhu, S Guo, S Niekum, D Ballard, and M Hayhoe. Human gaze assisted artificial intelligence: A review. In International Joint Conference on Artificial Intelligence, 2020. The dissertator is the first author, and takes the leading role in conceiving and designing the surveys and writing the papers.

have collected human gaze and action data in meal preparation [187], human-to-human (non-verbal) interactions [375], driving [228], and video game playing [362]. This allows researchers to study the relation between attention and decision. We hope that the Atari-HEAD dataset can serve a similar purpose for visual saliency and visuomotor behavior research.

The gaze data can be collected in parallel with actions. One concern with this approach is the hardware and software required to collect human gaze data. Recent progress in computer vision has improved eye tracker accuracy and portability by a significant margin. Appearance-based algorithms using convolutional neural networks have been shown to have better tracking accuracy and are more robust to visual appearance variations [368, 332, 169, 278, 369, 233], compared to more traditional approaches like hand-crafted feature-based or model-based algorithms. Advanced tracking software can estimate gaze in real-time from head poses and appearance without specialized hardware on low-cost devices such as webcams [232] and mobile tablets and phones [134, 169].

### **3.8.1.2 Related Work: Human Attention-Guided Imitation Learning**

With human gaze data, the learning problem for the agent is to learn the attention mechanism from humans in addition to learning a decision policy.

The first learning objective using human eye-tracking datasets could be training an agent to imitate human gaze behaviors, i.e., learning to attend

to certain features of a given image. The problem was formalized as a visual saliency prediction problem in computer vision research [144]. Recently this area has made tremendous progress due to deep learning as large-scale eye-tracking datasets became available for images [231, 186, 337, 51, 49, 169], videos [200, 322], and 360-degree videos [370, 338]. Visual saliency is a well-developed field in computer vision. We direct interested readers to recent review papers on the topics of saliency evaluation metrics [53], saliency model performance analyses [54, 119] and a closely related field called salient object detection [39].

Recent works have trained convolutional neural networks to accomplish the saliency prediction task [187, 362, 228, 76, 61]. A notable challenge here is *egocentric* gaze prediction in which the spatial distribution of the gaze is highly biased towards the image center, a problem further addressed by [228, 304].

In computer vision, traditional saliency prediction does not involve active tasks nor human decisions. The humans look at static images or videos in a free-viewing manner without performing any particular task and only the eye movements are recorded and modeled. Meanwhile, the aforementioned datasets all require humans to perform a task while collecting their gaze and action data. From a decision-learning perspective, human attention may provide additional information about their decisions, therefore it is intuitive to leverage learned attention models to guide the learning process of human decisions.

Similar to our results, experimental results have shown that including gaze information leads to higher accuracy in recognizing or predicting human actions, in reaching [245], human-to-human interaction [375], driving [336,

191, 62, 334], meal preparation [187, 275, 296, 136], and video game playing [359, 362].

Once the agent has learned both the attention and decision models from human data, it can perform the task on its own. It has been shown that incorporating a learned gaze model into imitation learning agents leads to a large performance increase, comparing to agents without attention information [362, 268, 61]. For real-world tasks like autonomous driving, it is reasonable to expect a similar improvement when incorporating human attention models. Due to physical constraints and safety reasons, this is yet to be explored but preliminary tests in simulated environments are possible.

### **3.8.1.3 Related Work: Human Attention in Robotics**

As robots, especially assistive robots, become more prevalent in our daily life, interaction and communication between robots and humans certainly have increased. Human-robot interaction (HRI) research aims to enhance such interaction and communication and shows that they can be facilitated by the sensitivity to human physiological signals, such as human gaze. We will review recent progress in robotics that utilizes human or robot gaze in HRI settings. For earlier work on this topic, we direct interested readers to two previous survey papers [261, 305].

Unlike vision, language, and decision learning tasks where gaze data is collected in advance, HRI requires robots to acquire human gaze during the interaction. In an ideal setting, a robot and its human partner are both

equipped with egocentric cameras, and the human is further equipped with an eye tracker. The robot has direct access to human camera and gaze data, from which it calculates the human's gaze vector in the robot's coordinate system [235]. Perhaps a more common but more challenging setting is that humans do not wear a camera nor an eye tracker, and the robot needs to estimate the human gaze vector by looking at their faces [7, 266]. A rough estimate can be computed from the human body and head orientation but this was shown to be much less informative than direct gaze measuring [229].

Once human gaze information is obtained, the next challenge is to interpret the meaning of the gaze. Social gaze between humans is relatively well studied, and a similar effort has been made for understanding human gaze when interacting with robots [248]. The interpretation of human gaze and its benefits are highly context-dependent. Humans and robots engage in various forms of interaction tasks. Similar to decision learning tasks discussed in the previous section, human gaze can facilitate robot learning during teaching [235, 267]. In a reversed setting, intelligent tutoring systems can monitor a human student's gaze to infer her mental or emotional state to encourage better engagement [150, 142]. Intention-revealing gaze enhances collaboration in object referring [86], teleoperation [344], shared autonomy [11], collaborative manipulation [133], and assisted reaching and grasping [274]. Human gaze can also help a robot infer the recipient of human verbal communication in a multi-party scenario [249].

#### 3.8.1.4 Related Work: Human versus Machine Attention

**Human vs. machine attention in vision and language tasks.** Human expert’s gaze is very efficient and accurate for solving vision tasks. The peak angular speed of the human eye during a saccade (fast, jumping eye movement) reaches up to 900 degrees per second [250]. This allows humans to move their foveae to the right place at the right time to attend to important features [78]. Therefore, human expert’s gaze serves as a good standard in many vision-related tasks for evaluating machine attention, or as a learning target for training machine attention [243, 352]. This approach is widely used in computer vision, see [220] for a review.

For example, visual saliency researchers train DNNs to predict human visual attention. One such paper has compared visual saliency models with human visual attention [171]. In vision-related language tasks, such as image captioning and visual question answering, it was found that the saliency maps of DNN models are different from human attention [69, 303, 120]. Understanding and quantifying such differences have provided insights on the performance, especially in failure scenarios, of these vision-language models.

**Visual explanation for deep RL.** Two classes of methods are widely used to generate visual interpretations of DNNs in the form of saliency maps: *gradient-based* and *perturbation-based*. Gradient-based methods compute saliency maps by estimating the input features’ influence on the output using the gradient information [283, 290, 197, 350, 277, 297, 272, 60, 371]. These

methods are for visualizing general DNNs but have been used to interpret deep RL agents [156, 327, 276, 151, 321]. We did not use gradient-based saliency maps for our analysis because they lack physical meaning and could be difficult to interpret.

Perturbation-based methods alter parts of the input image and measure how much the output is affected by the change. Hence there have been different methods of alternating the input [347, 90, 68, 374, 247]. These methods have been applied to Atari deep RL agents and can generate qualitatively meaningful saliency maps [107, 146, 108, 242]. However, without human attention as a reference, it is difficult to quantitatively analyze these saliency maps, which motivates our work.

Some methods change the architecture of the deep RL network by augmenting it with an explicit artificial attention module so that one can directly access its attention map [215, 341, 214]. Researchers have taken this approach and compared RL agents' attention with human attention [222]. However, these methods do not apply to general deep RL algorithms since they need to modify the original network architectures and retrain the new ones.

### **3.8.2 Future Work**

There are several promising future directions for research that arise from current progress.

**Future work for gaze and action prediction** We formulate the gaze prediction in Atari-HEAD as a saliency prediction problem, which ignores the sequential information in the eye movement data. One option would be using a recurrent neural network to represent information from past frames as memory, instead of stacking multiple images [208, 335, 348]. Recurrent network models also allow one to model eye movement scanpaths as a sequence prediction problem.

Deep neural networks are not directly interpretable in general, therefore our models provide little explanation on why a particular eye movement or an action was made. However, the modeling accuracy of gaze network and policy network indicates that human gaze and actions are indeed highly predictable in these games. Therefore our results could be treated as a tentative performance upper bound for traditional models that are interpretable but less accurate.

**Future Work for attention-guided imitation learning** A byproduct of the semi-frame-by-frame gameplay mode is human *option* [300]. We notice that human players often hold a key down until a sub-goal is reached, then release the key and plan for the next sequence of actions. This naturally segments the decision trajectories into temporally extended actions, i.e., options. It has yet to be explored whether a learning agent can learn from this type of human demonstrated options, but results from hierarchical imitation learning [175] indicate that this may indeed be possible.

**Future work for attention-guided reinforcement learning** *State representation learning through compression.* A good representation of the state space is critical for the learning agent’s performance, which should include all task-relevant features. However, too much information also poses a heavy burden on the learning process as indicated by the curse of dimensionality problem. [3, 182] proposed an algorithm that compresses the original state space based on rate-distortion theory [33]. The idea is to introduce an information bottleneck constraint on the learned state-space representation. Hypothetically, the human attention model can serve as an additional constraint that could help identify parts of the state space that should be preserved during the compression process. It would also be interesting to test whether humans and information bottleneck based compression algorithms agree on which visual features are important.

*Model learning.* In reinforcement learning, predicting the next frame based on the current frame and action has been used as an auxiliary learning task [148]. This prediction task is known as the action-conditional video prediction problem and is an interesting problem by itself [223]. Intuitively, not all pixels have the same importance given the task context. Attention should help re-weight the reconstruction loss that emphasizes important pixels around task-relevant objects.

*Feudal reinforcement learning.* [72, 316] is a hierarchical RL framework, in which a task manager learns to deliver high-level subgoals to a worker agent that learns to make low-level decisions. In a recent work that combines feudal

RL with deep neural networks, the manager was able to learn semantically meaningful subgoals [316]. One interesting comparison would be to test whether the feudal RL agent and the human players choose the same subgoal given an input state. Additionally, human attention information could serve as a supervised learning signal that could help the manager learn the subgoals.

**Future work for human versus machine attention** Multiple factors are important for interpreting our results and could explain the remaining differences between human attention and RL attention. The first one is the overt vs. covert attention issue discussed earlier. Humans store information in memory and do not need to constantly move their eyes to attend to all task-relevant objects. To complete the human attention map, in addition to gaze data, one will need to retrieve human covert attention from brain activity data, a technique that became possible recently [181, 67]. Another factor is human intrinsic reward. Humans are likely to have internal reward functions that are different from the ones provided by the game environment, and reward is known to affect attention [259, 181]. For now, given only the human gaze and overt attention model, RL agents with memory (e.g., LSTM agents) [107] may exhibit attention that is more human-like.

A closely related research direction compares a human player’s *policy* with an RL agent’s learned policy [212] which could further allow us to better understand the similarities and differences between humans and RL agents. However, as we have shown here, the difference in decisions could be due

to perception, which needs to be considered while comparing policies. Our approach lays the groundwork for future research in this direction.

### 3.9 Conclusion

How does the brain learn and make decisions to achieve behavioral goals in an information-rich environment, with limited cognitive resources?

At the second level of Marr’s paradigm, we emphasize the role of selective attention in human decision-making and learning. We collect a large-scale, high-quality dataset of human actions with simultaneously recorded eye movements while humans play Atari video games. The novel features of this dataset include human gaze data and a semi-frame-by-frame gameplay mode. The latter ensures that states and actions are matched, and allow enough decision time for human players.

We demonstrate the usefulness of the dataset through two modeling tasks: predicting human gaze and human actions. The scale and the quality of the dataset allow us to leverage deep neural networks to perform modeling, leading to promising results in both tasks. Moreover, adding human attention information into action modeling leads to a significant increase in model performance. We interpret these results as highlighting the importance of incorporating human visual attention in models of decision making and demonstrating the value of the current dataset to the research community.

How can we improve current artificial intelligence (AI) by studying these mechanisms of the brain, so that AIs can cope with the complexity in the real world?

The Atari-HEAD dataset addresses two major issues in IL and RL research; the first being reproducibility and the second being the need to bridge attention and control. We do this by providing human data that allows researchers to study how humans use visual attention to solve visuomotor tasks. We have shown promising results in saliency prediction, IL, and RL tasks using Atari-HEAD. The most exciting result is that the human attention model improves the performance of the decision-learning algorithm. We hope that the scale and quality of this dataset can provide more opportunities to researchers in the areas of visual attention, imitation learning, and reinforcement learning.

From our experiments and related studies, we conclude that there are at least five possible ways to incorporate attention information into deep IL/RL models [361, 352]:

1. Attention can be used as an additional channel of information, e.g., by concatenating a gaze map with the input image,
2. Attention can be used as a mask on input images or convolutional feature maps at intermediate layers of the deep network (e.g., AGIL). The mask can generate a representation of the image that highlights the attended visual features. This is the most common method so far.

3. Attention can be used as an auxiliary loss function that is added to the original learning objective (e.g., CGL).
4. Attention can be used to produce semantically meaningful samples in data augmentation (e.g., EXPAND).
5. Attention can be used as a pre-training stage for the further learning task. [74] has shown these are promising ways to speed up the learning process of an RL agent.

## Chapter 4

# The Modularization Hypothesis<sup>1</sup>

Let's revisit our primary research question: How does the brain learn and act in an information-rich environment, with limited cognitive resources, to achieve behavioral goals? Attentional control has provided one answer to this question. In this chapter, we continue our modeling effort at level II of Marr's paradigm. We hypothesize that the brain uses a divide-and-conquer strategy called the *modularization hypothesis*: A complex behavior goal can be broken down into multiple subgoals, each of which requires specific visual information, has different objectives (in terms of rewards), and prefers different actions [289, 259, 307, 24]. As introduced earlier, a theoretical basis for modeling reward-seeking behavior is the Markov decision process (MDP) and reinforcement learning (RL). In this chapter, we formalize the modularization hypothesis in the context of MDP and RL, and show how this hypothesis would help us model and predict human behaviors in natural tasks.

---

<sup>1</sup>This chapter of work is based on the following publication: Ruohan Zhang, Shun Zhang, Matthew H Tong, Yuchen Cui, Constantin A Rothkopf, Dana H Ballard, and Mary M Hayhoe. Modeling sensorymotor decisions in natural behavior. PLoS computational biology, 14(10):e1006518, 2018. The dissertator is the first author, and takes the leading role in conceiving and designing the analysis, contributing analysis tools, performing the analysis, and writing the paper.

Modeling and predicting visually guided behavior in humans is challenging. In various contexts, it is unclear what information is being acquired and how it is being used to control behaviors. An empirical investigation of natural behavior has been limited, largely because it requires immersion in natural environments and monitoring of ongoing behavior. However, recent technical developments have allowed more extensive investigation of visually guided behavior in natural contexts [118].

A theoretical basis for modeling such behavioral sequences is reinforcement learning (RL). Since the breakthrough work by [298], a rapidly increasing number of studies have used a formal reinforcement learning framework to model reward-seeking behaviors. Numerous studies have linked sensory-motor decisions to the underlying dopaminergic reward machinery [329, 118]. The basic mechanisms of reinforcement learning, such as reward estimation, temporal-difference error, model-free and model-based learning, and discount factor, have been linked to a broad range of brain regions [115, 130, 162, 91, 178, 55, 70, 210, 80, 301]. Because studies of the neural circuitry involve very restrictive behavioral paradigms, it is not known how these effects play out in the context of natural visually guided behavior. Similarly, the application of RL models to human behavior has been restricted almost exclusively to simple laboratory paradigms, and there are few formal attempts to model natural behaviors [117]. The goal of the presented work is to predict action choices in a virtual walking setting by estimating the subjective value of some of the sub-tasks that the sensory-motor system must

perform in this context. We show that it is possible to estimate the subjective reward values of behaviors such as obstacle avoidance and path following, and accurately predict the trajectories walkers take through the environment. This demonstration suggests a potential analytical tool for the exploration of natural behavioral sequences.

**Modular reinforcement learning** An important factor that makes standard RL difficult in modeling natural behaviors is its sophistication and resulting computational burden as a model for general reward-seeking behaviors. The natural environment has at least two features that could make RL/IRL algorithms computationally intractable. First, a large number of task-relevant objects may be present, hence the decision state space is likely to be high-dimensional. Standard RL suffers from the *curse of dimensionality* with high-dimensional state space, where the computational burden grows exponentially with the number of state variables [298, 258]. Second, the natural environment is ever-changing such that humans must make decisions under different situations although these situations might have similar components. Living in a natural environment requires a decision-maker to be able to *transfer* knowledge learned from previous experience to a new situation. In contrast, an RL agent is often trained and tested repeatedly in a fixed environment. The optimal behavior is obtained through either a model-based dynamic programming approach that requires full knowledge of the environment, or a model-free learning approach that requires a large amount of experience. Both approaches generally put

a heavy burden on memory storage or computation to calculate the optimal behavior. Consequently, both of them may not be suitable for the real-time decision-making strategy in natural conditions since decision-makers encounter new environments all the time and need to make decisions with reasonable cognitive load. For these reasons, standard RL must be extended to make computation tractable.

An extension of standard RL named *modular* reinforcement learning utilizes divide-and-conquer as an approximation strategy [258, 265, 288]. The modular RL takes the statistical structure present in the environment, decomposes a task into *modules* where each module solves a subgoal of the original task. Generally, an arbitrator is required to synthesize module policies and make final decisions. Modularization alleviates the problem of the curse of dimensionality since each module only concerns a subset of state variables. Introducing a new state variable may not affect the entire state space and cause its size to grow exponentially. Additionally, the decomposition naturally allows the decision-maker to learn a behavior specifically for a module and reuse it later in a new environment. Under the modular RL framework, a more sample-efficient IRL algorithm is possible [258], which matters for modeling natural human behaviors since such behavioral data is often expensive to collect.

**Estimating the discount factor** A frequently overlooked variable in RL is the discount factor that determines how much a decision-maker weighs future

reward compared to immediate reward. In the agent-environment interaction paradigm, a standard RL model typically treats the discount factor as a part of the environment and as fixed. The alternative approach is to view the discount factor as a subjective decision-making variable that is part of the agent and may vary. Behavioral neuroscience studies suggest that the magnitude of the discount factor is correlated with serotonin level in human subjects [270]. As a consequence decision-makers may exhibit between-subject variations [293].

In the multiprocessing or multitasking context, discount factor plays another important role: The same decision-maker may use different discount factors for different tasks. An fMRI study by [301] suggests that different cortico-basal ganglia loops are responsible for reward prediction at different time scales, allowing multiple discount factors to be implemented. Hence it is necessary to extend the standard RL model to adapt discount factors to different human subjects and tasks. A modular approach is ideal for this modeling effort. Allowing different modules to have their own discount factors makes the model flexible in modeling potential variations in human data.

## 4.1 Summary of Contributions

1. Behaviors can be decomposed into multiple modules. Each module is modeled by a Markov decision process. Modules are combined using the modular reinforcement learning framework and executed in parallel (4.2.1).

2. Modular inverse reinforcement learning (4.2.2) can estimate module rewards and discount factors from behavioral data with high sample efficiency.
3. The proposed algorithms are first validated in simulation (4.3). A discussion of how modules coordinate to make decisions is also included (4.3.3).
4. Estimated rewards and discount factors from human data can be used to train a modular RL agent to navigate like a human (4.4).
5. Simulated robots can use this model to infer human navigation behaviors and navigate safely among humans (4.5).

## 4.2 Modular Reinforcement Learning and Inverse Reinforcement Learning

### 4.2.1 Modular Reinforcement Learning

The modular reinforcement learning utilizes divide-and-conquer as an approximation strategy [258, 265, 288]. Formally, a *module* is a subtask of the original task. Each module is hence a simpler problem so that its value function and policy can be learned or calculated efficiently. A module is also modeled by an MDP  $\langle \mathcal{S}^{(n)}, \mathcal{A}, \mathcal{P}^{(n)}, \mathcal{R}^{(n)}, \gamma^{(n)} \rangle$ , where  $n$  is the index of the  $n$ th module. Note that each module has its own state space, transition function, reward function, and discount factor, but the action space is shared between modules because all modules reside in a single agent.

Let  $N$  be the number of modules and  $Q^{(n)\pi^{(n)}}$  denote module Q-value function (Eq. 2.2) of the  $n$ th module conditioned on module policy  $\pi^{(n)}$ . For simplicity, we will drop  $\pi^{(n)}$  and write  $Q^{(n)}$ . Let  $Q$  without superscription denote the global Q function (also drop global policy  $\pi$ ). Modular RL sums module Q functions to obtain the global Q function [263, 288]:

$$Q(s, a) = \sum_{n=1}^N Q^{(n)}(s^{(n)}, a) \quad (4.1)$$

There can be multiple *module objects* of a module, e.g., several identical obstacles nearby to avoid. The number of objects of each module is denoted as  $M^{(1)}, \dots, M^{(N)}$ . Note that for a given module, its module objects share the same  $Q^{(n)}$  since their module MDPs are identical. But at a given time they could be in different states relative to the agent’s reference frame which can be denoted as  $s^{(n,m)}$  for module  $n$  object  $m$ . To generalize the above equation:

$$Q(s, a) = \sum_{n=1}^N \sum_{m=1}^{M^{(n)}} Q^{(n)}(s^{(n,m)}, a) \quad (4.2)$$

This assumes independent transition functions between module objects [258]. A module action-value function  $Q^{(n)}$  may be calculated from solving Bellman equations using dynamic programming or through standard learning algorithms with enough experience data, which we argue to be infeasible for humans performing natural tasks.  $Q^{(n)}$  needs to be calculated efficiently with a reasonable cognitive load.

In certain tasks, both the state transition function and reward function are deterministic hence the expectation in Eq (2.2) can be dropped. Since

each module Q function only considers a single source of reward from a single module object, and assuming a policy that leads the agent directly to the module object,  $Q^{(n)}(s^{(n,m)}, a)$  takes the following simple form:

$$Q^{(n)}(s^{(n,m)}, a) = r^{(n)}(\gamma^{(n)})^{d(s^{(n,m)}, a)} \quad (4.3)$$

where  $r^{(n)}$  is the reward for the  $n$ th module,  $\gamma^{(n)}$  is its discount factor, and  $d(s^{(n,m)}, a)$  is the spatial or temporal distance between the agent and the module object  $m$  after taking action  $a$  at state  $s^{(n,m)}$ . Note Eq (4.3) converts value function back to its simplest form in [80]. This simple form allows a decision-maker to calculate the action-value for a state efficiently when needed instead of beforehand. This matters when humans need to make decisions fast and when it is computationally expensive to calculate value functions using a standard RL algorithm. It is also unlikely for a human to pre-compute the values for all future states and use dynamic programming to obtain a global policy when they visit the environment for the first time. Doing so would at least require a human to store Q-values for relevant states (a Q-table) in its memory system, which is convenient for an artificial agent but would be difficult for a real-time human decision-maker.

Why does modular RL alleviate the problem of curse of dimensionality? Consider the joint state space of a standard RL which can be represented as the Cartesian product of the module state spaces:  $\mathcal{S} = \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \dots$ . The computation cost for one iteration in value iteration (a popular RL algorithm) is  $O(|\mathcal{S}|^2|\mathcal{A}|)$  where  $|\cdot|$  denotes the cardinality of a set [160]. When a new module

$\mathcal{S}^{(N)}$  is added, the cost of standard RL becomes  $O(|\mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \dots \times \mathcal{S}^{(N)}|^2 |\mathcal{A}|)$ , while the cost of modular RL becomes  $O(|\mathcal{S}^{(1)}|^2 |\mathcal{A}|) + O(|\mathcal{S}^{(2)}|^2 |\mathcal{A}|) + \dots + O(|\mathcal{S}^{(N)}|^2 |\mathcal{A}|)$ . Therefore the computational cost increases additively in modular RL instead of multiplicatively.

**Visualizing modular reinforcement learning** Eq (4.3) bridges modular RL with an important planning method called artificial potential field [163, 10, 135]. Similar to a potential field, we use a value surface to visualize the value function. Each module object is associated with a value surface. The module reward controls the maximum absolute height of the surface, and the discount factor controls temporal or spatial discounting rates. Module value surfaces can be composed directly by summation or integration to produce a multi-module value surface. The concept of value surfaces and their combination is illustrated in Fig 4.1. Given a composed value surface as in Fig 4.1f, a modular RL agent would choose actions that lead to a local minimum on the surface. A sequence of actions could construct a trajectory in Fig 4.2a which traverses through a sequence of local minima.

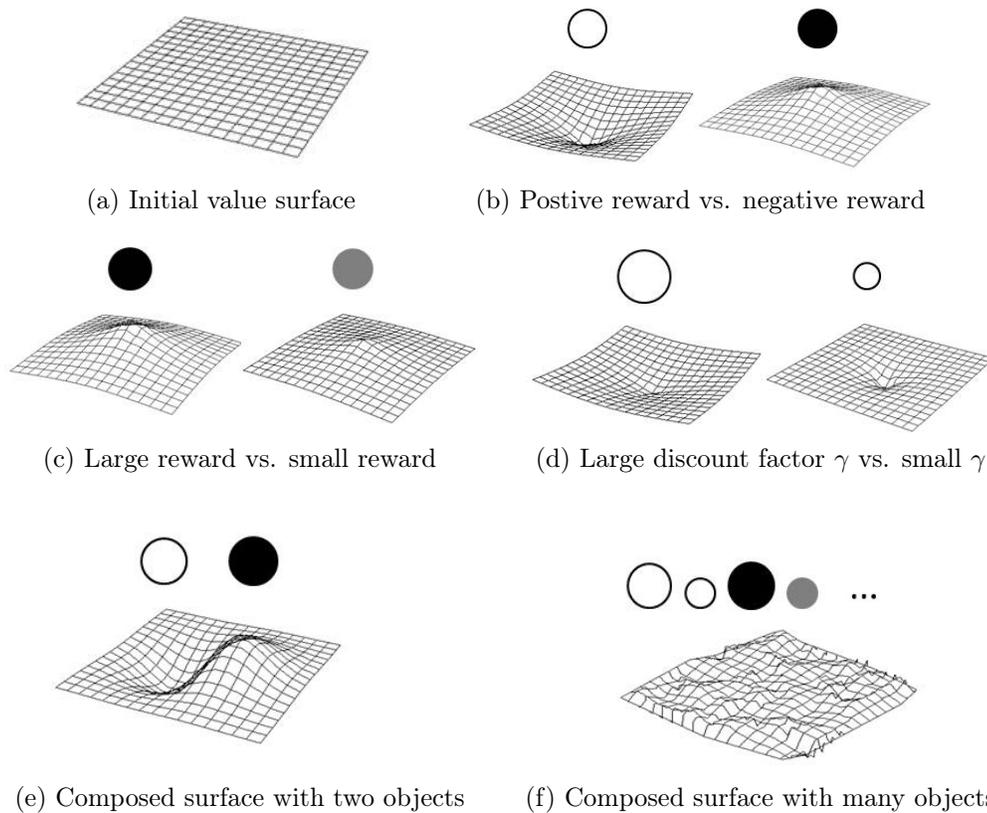


Figure 4.1: The concept of modular reinforcement learning illustrated using value surfaces. (a) The value surface is flat without any reward signal. (b) A module object with positive reward has positive weight, and one with negative reward has negative weight. They bend the value surface to have negative and positive curvatures respectively. Therefore, an agent desires to follow the steepest descent to minimize energy, or equivalently, to maximize reward. (c) An object with larger weight bends the surface more. (d) An object with greater discount factor  $\gamma$  has larger influence over distance. (e,f) Composing different objects with different rewards and  $\gamma$ s results complicated value surfaces that can model an agent's value function over the entire state space.

### 4.2.2 Modular Inverse Reinforcement Learning

While reinforcement learning aims at finding the optimal policy given a reward function, inverse reinforcement learning (IRL) attempts to infer the unknown reward function given the agent behavioral data in the form of state-action pairs  $(s_t, a_t)$  [219, 2, 372, 244]. Our work is largely based on the modular IRL algorithm by [258] which pioneered the first modular IRL algorithm. Given the modular RL formulation in the previous section, the goal of modular IRL is to estimate the underlying reward and discount factor for each module to recover the value function, given a sequence of observed state-action pairs, i.e., a trajectory that traverses through the state space, as shown in Fig 4.2a.



Figure 4.2: Maximum likelihood modular inverse reinforcement learning. (a) From an observed trajectory (a sequence of state-action pairs), the goal of modular IRL is to recover the underlying value surface. (b) Maximum likelihood IRL assumes that the probability of observing a particular action (red) in a state is proportional to its  $Q$ -value among all possible actions as in Eq (4.4).

We follow the Bayesian formulation of IRL [244, 194], Maximum Likelihood IRL [14], and improve the modular IRL algorithm in [258]. These approaches assume that the higher the  $Q$ -value for action  $a_t$  in state  $s_t$ , the

more likely action  $a_t$  is observed in behavioral data. Let  $\eta$  denote the confidence level in optimality (the extent to which an agent selects actions greedily, default to be 1), and let  $\exp(\cdot)$  denote the exponential function. The likelihood of observing a certain state-action pair is modeled by the softmax function with Gibbs (Boltzmann) distribution, as illustrated in Fig 4.2b:

$$P(a_t|s_t, Q, \eta) = \frac{\exp(\eta Q(s_t, a_t))}{\sum_{a \in \mathcal{A}} \exp(\eta Q(s_t, a))} \quad (4.4)$$

Let  $T$  denote the total length of the trajectory. The overall likelihood  $\mathcal{L}$  for observed data  $D = \{(s_1, a_1), \dots, (s_T, a_T)\}$  is the product of the likelihood of individual state-action pairs, given the states are Markovian and action decisions are independent:

$$\mathcal{L} = P(D|Q, \eta) = \prod_{t=1}^T \frac{\exp(\eta Q(s_t, a_t))}{\sum_{a \in \mathcal{A}} \exp(\eta Q(s_t, a))} \quad (4.5)$$

Next, the global action-value function  $Q(s_t, a_t)$  is decomposed using Eq (4.2) with module Q functions  $Q^{(1:N)}$ , therefore the likelihood becomes:

$$\begin{aligned} \mathcal{L} &= P(D|Q^{(1:N)}, \eta) \\ &= \prod_{t=1}^T \frac{\prod_{n=1}^N \prod_{m=1}^{M_t^{(n)}} \exp(\eta Q^{(n)}(s_t^{(n,m)}, a_t))}{\sum_{a \in \mathcal{A}} \prod_{n=1}^N \prod_{m=1}^{M_t^{(n)}} \exp(\eta Q^{(n)}(s_t^{(n,m)}, a))} \end{aligned} \quad (4.6)$$

Take the log of the likelihood function:

$$\begin{aligned} \log \mathcal{L} &= \sum_{t=1}^T \left( \sum_{n=1}^N \sum_{m=1}^{M_t^{(n)}} \eta Q^{(n)}(s_t^{(n,m)}, a_t) \right. \\ &\quad \left. - \log \sum_{a \in \mathcal{A}} \prod_{n=1}^N \prod_{m=1}^{M_t^{(n)}} \exp(\eta Q^{(n)}(s_t^{(n,m)}, a)) \right) \end{aligned} \quad (4.7)$$

Substituting Eq (4.3) into Eq (4.7):

$$\begin{aligned} \log \mathcal{L} &= \sum_{t=1}^T \left( \sum_{n=1}^N \sum_{m=1}^{M_t^{(n)}} \eta r^{(n)}(\gamma^{(n)})^{d(s_t^{(n,m)}, a_t)} \right. \\ &\quad \left. - \log \sum_{a \in \mathcal{A}} \prod_{n=1}^N \prod_{m=1}^{M_t^{(n)}} \exp(\eta r^{(n)}(\gamma^{(n)})^{d(s_t^{(n,m)}, a)}) \right) \end{aligned} \quad (4.8)$$

The variables to be estimated from the data are module rewards  $r^{(1:N)}$  and discount factors  $\gamma^{(1:N)}$ . The number of modules  $N$ , the number of objects for each module  $M_t^{(1)}, \dots, M_t^{(N)}$ , and distances  $d(s_t^{(n,m)}, a_t)$  for each object are all state information and can be observed from the environment. This formulation follows closely the work by [258], extending it to use the new formulation of modular RL, handle multiple objects of each module, estimate the discount factors, and derive a slightly different objective function.

#### 4.2.2.1 Sparse Modular Inverse Reinforcement Learning

Modular IRL can only guess which objects are actually being considered by the decision-maker when chosen an action. To address this problem, we can further add a  $L_1$  regularizer  $-\lambda \sum_{n=1}^N \|r^{(n)}\|_1$  to Eq (4.8), which causes some

module rewards to become 0 so these modules would be ignored in decision making. This is an extension of using a Laplacian prior in Bayesian IRL [244]. In addition to the benefit from an optimization perspective, the regularization term has the following important interpretation in terms of explaining natural behaviors.

A *hypothetical module set* is a set  $\mathcal{H} = \{1, \dots, N\}$  contains  $N$  modules that could potentially be of an agent's interest. However, due to the limitations in computational resource, the agent can only consider a subset of  $\mathcal{H}$  at a time, denoted  $\mathcal{H}'$ . In a rich environment many modules' rewards would be effectively zero at current decision step, hence  $|\mathcal{H}'| \ll |\mathcal{H}|$ . For instance, a driving environment could contain hundreds of objects in  $\mathcal{H}$ . But a driver may pay attention to only a few. The regularization constant  $\lambda$  serves as a cognitive capacity factor that helps determine  $\mathcal{H}'$  from the observed behaviors. Therefore the final objective function of modular IRL is:

$$\begin{aligned}
& \max_{r^{(1:N)}, \gamma^{(1:N)}} \sum_{t=1}^T \left( \sum_{n=1}^N \sum_{m=1}^{M_t^{(n)}} \eta r^{(n)} (\gamma^{(n)})^{d(s_t^{(n,m)}, a_t)} \right. \\
& \left. - \log \sum_{a \in \mathcal{A}} \prod_{n=1}^N \prod_{m=1}^{M_t^{(n)}} \exp(\eta r^{(n)} (\gamma^{(n)})^{d(s_t^{(n,m)}, a)}) \right) \\
& - \lambda \sum_{n=1}^N \|r^{(n)}\|_1 \\
& s.t. \ 0 \leq \gamma^{(n)} < 1.
\end{aligned} \tag{4.9}$$

Sometimes the sign for  $r^{(n)}$  is easy to determine beforehand, since it is straight-

forward to tell whether a module has a positive or negative reward. Note that if we are to fit  $r^{(1:N)}$  and  $\gamma^{(1:N)}$  simultaneously, the above objective function is non-convex. However, the objective becomes convex if only fitting  $r^{(1:N)}$ . Since  $\gamma^{(n)}$  is in range  $[0, 1)$ , one can perform a grid search over values for  $\gamma^{(1:N)}$  with step size  $\epsilon$  and fit  $r^{(1:N)}$  at each possible  $\gamma^{(1:N)}$  value. This allows us to find a solution within  $\epsilon$ -precision of the true global optimum. Alternatively, various non-convex optimization techniques such as the differential evolution techniques [292] can also be used.

### 4.3 Simulation Results

As a sanity check, computer simulations in an artificial multitask navigation environment were performed as an evaluation of the proposed algorithms. The environment is a 2D gridworld that resembles the virtual room we use for the human experiments which will be introduced later. The validity of the modular IRL is proved empirically by showing its ability to recover true module rewards and discount factors with high accuracy given enough behavioral data. Meanwhile, it requires significantly fewer data samples to obtain high prediction accuracy compared to a standard Bayesian IRL algorithm [244], presumably because the state space is reduced significantly by modularization. Sparse modular IRL is shown to further improve sample efficiency if task-irrelevant modules are present. Unlike computer simulated experiments in which one can easily generate millions of behavioral data, human experiments have a more expensive data collection procedure in general. Therefore sample efficiency of

sparse modular IRL is an important advantage in modeling natural human behaviors, which will be seen in the result section.

### 4.3.1 Simulation: 2D Gridworld Navigation

Using a canonical 2D gridworld in standard RL research, the goals are to validate that the modular IRL algorithm can correctly estimate the rewards and discount factors, to demonstrate its advantage over the previous method, and to show an application of the sparse modular IRL. A portion of the gridworld is shown in Fig. 4.3. Different module objects are indicated by different colors and shapes. Behavior data (state-action pair samples) are collected from a modular RL agent.

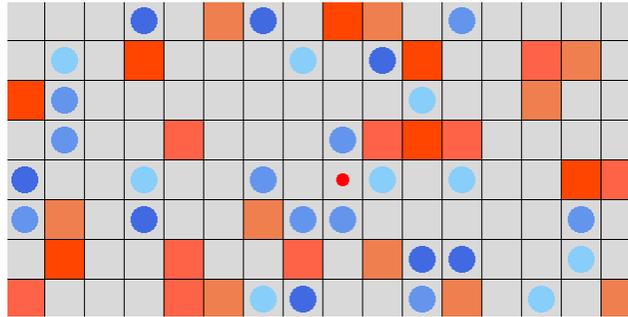


Figure 4.3: Part of the 2D gridworld test domain. Red squares are obstacles with negative rewards. Blue circles are targets with positive rewards. The small red dot is the modular RL agent. Different colors indicate different modules with distinct rewards and discount factors. The objects of the same module have the same color.

We first demonstrate that modular IRL can recover module rewards and discount factors correctly. The environment contains six modules each with ten objects. Three of them have positive rewards and the other three have

negative rewards. 10 grid worlds are generated with random layouts of objects. The agent navigates each world for 6,000 steps. Eq. (4.8) is used to estimate  $r^{(1:6)}$  and  $\gamma^{(1:6)}$  and we calculate the mean estimation and standard deviation. The results are shown in Table 4.1, it is evident that our algorithm is highly accurate in recovering rewards and discount factors given a large amount of data as in this experiment.

	$r^{(1)}$	$r^{(2)}$	$r^{(3)}$
Truth	+5	+10	+15
Estimation	+5.00±0.02	+9.94±0.03	+15.02±0.03
	$r^{(4)}$	$r^{(5)}$	$r^{(6)}$
Truth	-5	-10	-15
Estimation	-4.97±0.02	-10.03±0.03	-14.85±0.07
	$\gamma^{(1)}$	$\gamma^{(2)}$	$\gamma^{(3)}$
Truth	0.7	0.6	0.5
Estimation	0.70±0.00	0.60±0.00	0.500±0.00
	$\gamma^{(4)}$	$\gamma^{(5)}$	$\gamma^{(6)}$
Truth	0.3	0.2	0.1
Estimation	0.30±0.00	0.20±0.00	0.10±0.00

Table 4.1: Estimated rewards and discount factors comparing to the ground truth for the six modules in the 2D gridworld experiment. The results are presented as mean  $\pm$  standard error. The estimations are highly accurate due to the availability of a large amount of data.

#### 4.3.1.1 Modular vs. Standard Inverse Reinforcement Learning

In modeling natural behaviors, one particularly important aspect of a machine learning algorithm is its sample efficiency, given that it could be expensive to collect behavior data unlike in computer simulation. The performance of modular IRL on sample efficiency is compared with a standard

non-modular Bayesian IRL [244]. We use a Laplacian prior in Bayesian IRL since the rewards are sparse. Fig. 4.4a shows the results. The test environment has 4 modules and each has 4 objects. Both algorithms are tested with a different number of samples (state-action pairs) and then compare the policies generated using the learned rewards. Policy agreement is defined as the proportion of the states that have the same policy as the ground truth, which is used because the outputs of these two algorithms are weights and rewards that can not be directly compared. Modular IRL obtained nearly 100% policy agreement with far fewer samples compared to the Bayesian IRL.

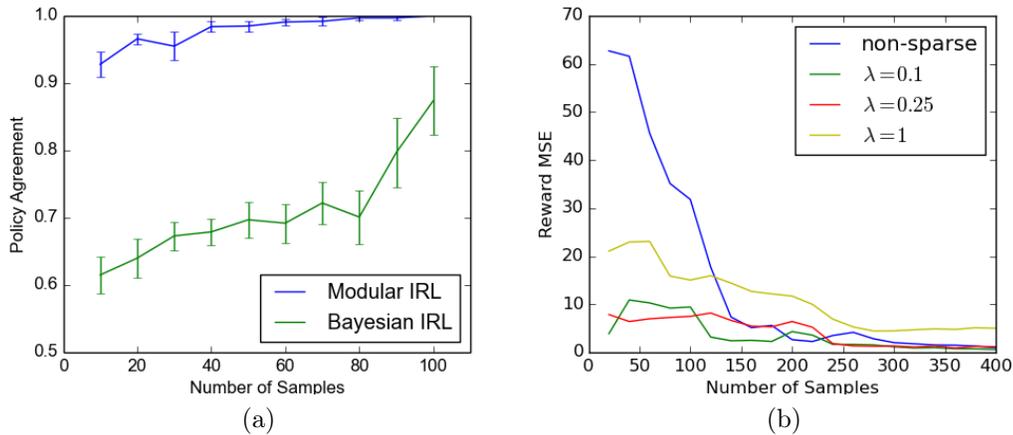


Figure 4.4: (a) Modular IRL vs Bayesian IRL on sample efficiency, measured by policy agreement  $\pm$  standard error ( $N = 10$ ). Modular IRL has significant higher sample efficiency. (b) Modular IRL vs sparse modular IRL on sample efficiency, measured by mean squared error (MSE) of estimated reward. Sparsity can greatly improve sample efficiency with a carefully chosen value of  $\lambda$ .

### 4.3.1.2 Sparse Modular Inverse Reinforcement Learning

Next we evaluate the performance of sparse modular IRL algorithm as in Eq. (4.9) on sample efficiency. The gridworld contains 10 modules and each has 10 objects. The agent has limited attention so it only considers 2 modules, i.e., the agent makes decisions by treating all other modules to have zero rewards. Therefore, the hypothetical module set has size  $|\mathcal{H}| = 10$  and actual module set  $|\mathcal{H}'| = 2$ . We use Eq. (4.9) to recover  $r$  and  $\gamma$ .

The mean squared error (MSE) of the estimated reward is shown in Fig. 4.4b. If data is scarce, the sparse version of the modular IRL algorithm ( $\lambda = 0.1, 0.25$ ) can recover rewards more accurately than the non-sparse version. Sparse modular IRL correctly identifies modules that the agent paid attention to, indicated by low MSE values obtained. As the regularization constant  $\lambda$  controls the importance of the regularization term, a very large  $\lambda$  introduces too much bias in estimation and may fail to converge to the truth, as shown by  $\lambda = 1$ . One can use standard cross-validation techniques in choosing the value for  $\lambda$ .

Unlike computer simulated experiments where one can easily generate millions of samples, human experiments generally require a more expensive data collection process. Therefore the sample efficiency property of modular RL and IRL is an important advantage in modeling natural human behaviors.

### 4.3.2 Simulation: Driving

The goal of the second simulation is to demonstrate that the modular IRL can model the individual differences in agent behavior with rewards and discount factors. We implement another canonical test environment in RL, namely the simulated 3-lane driving task from [2], as shown in Fig. 4.5. The setup follows the original paper, except that the actions are to drive to the left/right lane or stay in the current lane. The driving task can be naturally

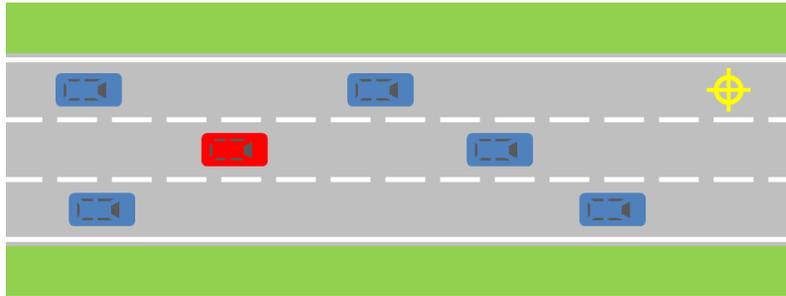


Figure 4.5: The simulated driving environment. Red car: the modular RL agent. Blue cars: other cars on the road. Yellow cross: the current target position for the agent.

decomposed into three modules: avoiding other cars, staying on the road, and driving to the target positions. The states are the distances to the module objects. We design a “nice” driving agent (receives a penalty for hitting other cars) and an “aggressive” agent (receives a positive reward for hitting other cars). Each agent drives for five trials with each trial consists of 3,000 data samples. Similar to the gridworld experiment, Eq. (4.8) is used to estimate the rewards and discount factors. The results are shown in Table 4.2. The modular IRL algorithm is accurate in recovering the rewards and discount factors for

both agents. This indicates that modular IRL could be used as a model for explaining various behaviors in the same environment, due to the differences in underlying rewards and discount factors. This issue will be explored more in section 4.4.2, in which actual human navigation data is modeled.

	Nice	Driver	
	$r^{(car)}$	$r^{(road)}$	$r^{(target)}$
Truth	-10	-2	+5
Est.	$-11.17 \pm 0.74$	$-2.02 \pm 0.12$	$+5.22 \pm 0.52$
	$\gamma^{(car)}$	$\gamma^{(road)}$	$\gamma^{(target)}$
Truth	0.2	0.1	0.8
Est.	$0.18 \pm 0.02$	$0.10 \pm 0.02$	$0.81 \pm 0.01$
	Aggressive	Driver	
	$r^{(car)}$	$r^{(road)}$	$r^{(target)}$
Truth	+5	-2	+10
Est.	$+5.04 \pm 0.13$	$-2.01 \pm 0.13$	$+9.52 \pm 0.76$
	$\gamma^{(car)}$	$\gamma^{(road)}$	$\gamma^{(target)}$
Truth	0.4	0.1	0.9
Est.	$0.40 \pm 0.01$	$0.10 \pm 0.05$	$0.89 \pm 0.02$

Table 4.2: Estimated rewards and discount factors for the car, road, and target modules, for the nice driver and the aggressive driver. The results are presented as mean  $\pm$  standard deviation.

### 4.3.3 Action Selection in Modular Reinforcement Learning<sup>2</sup>

Modular RL introduces a new challenge: each module has its own reward function, policy, and preferred action in each state, but the action space is

<sup>2</sup>This section of work is based on the following publication: Ruohan Zhang, Zhao Song, and Dana H Ballard. Global policy construction in modular reinforcement learning. In AAAI, pages 4226–4227, 2015. The dissertator is the first author, and takes the leading role in conceiving and designing the analysis, collecting the data, contributing analysis tools, performing the analysis, and writing the paper.

shared among modules, hence a global coordinator needs to choose an action to resolve the conflicts between modules. A possible but expensive approach is to use a weighted outcome of module policies to determine the global policy, and these weights can be learned [139, 288]. In practice, heuristics are often used to avoid the additional cost of learning the weights and the global policy. For example, Modular RL sums module Q functions to obtain the global Q function (Eq. 4.1), which maximizes the collective utility of all modules and will be referred to as the *Module Aggregation* method. However, there are other possible useful heuristics.

Let  $W_i(s_i)$  denote the weight of module  $i$  at state  $s_i$ . We propose two additional simple heuristics [360, 353]:

- *Module Selection* maximizes utility of the module with the highest weight. Select module  $i^* = \arg \max_i W_i(s_i)$ , and choose action  $a^* = a_i$ .
- *Module Voting* uses shareholder voting heuristic. We let each module to vote for its optimal action, instead of one-module-one-vote, the vote is weighted. Let  $K(s_i, a)$  be the vote count for global action  $a$ , then  $K(s_i, a) = \sum_i W_i(s_i)$  for all module  $i$  whose optimal action  $a_i = a$ . This is saying that each module  $i$  put  $W_i(s_i)$  votes on its optimal action  $a_i$ . Global action is selected as the action with highest number of votes,  $a^* = \arg \max_a K(s_i, a)$ .

The key is to choose the weight  $W_i$  which ideally should encode the importance, as well as the "urgency" of each module in the current state. As a

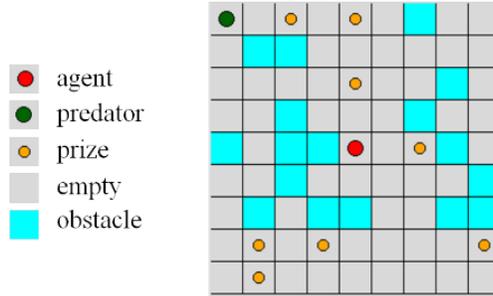


Figure 4.6: Test domain for action selection in modular RL.  $R_{prize} = +10$ ;  $R_{obstacle} = -10$ ;  $R_{predator} = -100$ .  $\gamma_{prize} = 0.7$ ;  $\gamma_{obstacle} = 0$ ;  $\gamma_{predator} = 0.1$ .

preliminary attempt, we choose  $W_i(s_i) = \sigma(Q_i(s_i, a))$ , where  $\sigma(Q_i(s_i, a))$  is standard deviation of Q values across actions for the module  $i$ . Intuitively, the standard deviation measures the variance of expected return when taking different actions in the current state, indicating how important this state is for a module to take control. Alternatively, some type of regret [299] could also be used in determining the weight.

To test these heuristics, we use a 9x9 grid-world shown in Fig. 4.6, similar to the previous ones. Our agent starts at the center. The dark dot is a predator, starting at the upper left corner of the map, which chases the agent with a probability of 0.5 and chooses a random action otherwise. Being captured by the predator resulted in the termination of an experiment trial and a large negative reward. A trial is successful if the agent collects all prizes within 250 steps, without being captured by the predator. We train each type of module using Sarsa( $\lambda$ ) with replacing traces [298].

Two performance criteria are the average success rate and the average

number of steps to complete a successful trial. We randomly pick 10% of cells to contain a prize. Let  $p_{obstacle}$  denote the proportion of cells being obstacle. Since this value defines task difficulty, we choose  $p_{obstacle} \in [0, .2]$  with step size of 0.01, resulted in 21 levels of difficulty. For each level, we randomly generate  $10^3$  maps with different layouts, and the agent navigates each map for 5 trials, testing one algorithm per trial. For comparison, we also show the performance of two baseline algorithms: a random agent and a reflex agent. The reflex agent chooses the action which maximizes its one-step-look-ahead reward. The results are shown in Fig. 4.7. At least for this simple domain, three heuristics have very similar performance. It would be interesting to test which strategy is actually used by the biological brain to make decisions (assuming the underlying mechanism is modular RL). The initial guess is that the module selection strategy has the lowest cognitive load. This is supported by previous research using gaze data to show that there might be only one module being activated for decision making at a time, meanwhile, other modules still update their state information but do not participate in making the decision [155].

## 4.4 Human Navigation Experiment in Virtual Reality

### 4.4.1 Experiment Design

Spatial navigation has been used as a canonical benchmark task for standard RL/IRL algorithms in machine learning, and therefore is selected as the experimental domain for testing our model [366, 367]. The task is an ideal testbed for modular RL since it is convenient for introducing multiple

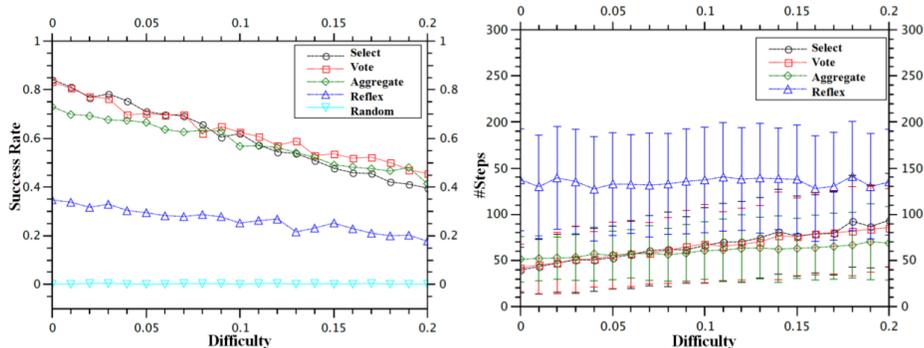


Figure 4.7: Left: Success rate. Right: Number of steps to complete a successful trial. The action selection strategies we proposed have similar performance in the gridworld.

(sub-)tasks.

Virtual reality (VR) and motion tracking were employed to create a naturalistic environment with a rich stimulus array while maintaining experimental control. Fig 4.8 shows the basic setup. The subject wore a binocular head-mounted display (the nVisor SX111 by NVIS) that showed a virtual room ( $8.5 \times 7.3$  meters). The subject's eye, head, and body motion were tracked while walking through the virtual room. Subjects were recruited from a subject pool of undergraduates at the University of Texas at Austin, and were naive to the nature of the experiment. The human subject research is approved by the University of Texas at Austin Institutional Review Board with approval number 2006-06-0085 [307].

Although we do not know the set of normal subtasks involved in walking through a room like this, three plausible candidates might be following a path across the room, avoiding obstacles, and perhaps heading towards target

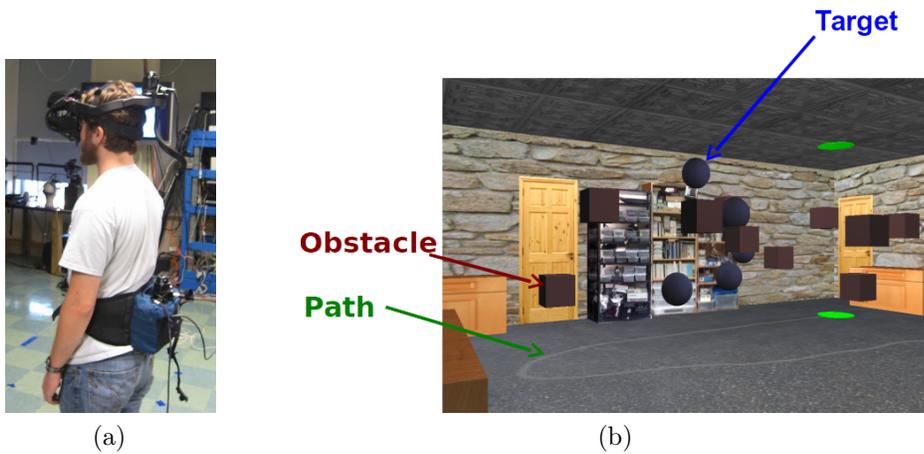


Figure 4.8: The virtual-reality human navigation experiment with motion tracking. (a) A human subject wears a head mounted display (HMD) and trackers for eyes, head, and body. (b) The virtual environment as seen through the HMD. The red cubes are obstacles and the blue spheres are targets. There is also a gray path on the ground leading to a goal (the green disk). At the green disk the subject is ‘transported’ to a new ‘level’ in a virtual elevator for another trial with a different arrangement of objects.

objects. To capture some of this natural behavior we asked subjects to collect the targets (blue spheres) by intercepting them, follow the path (the gray line), and/or avoid the obstacles (red cubes). Objects disappeared after a collision. This type of state transition function encourages subjects to navigate through the virtual room instead of sticking at a single target.

The global task has at least three *modules*: following the path, collecting targets, and avoiding obstacles. We gave subjects four types of instructions that attempt to manipulate their reward functions (and potentially the discount factors), resulting in four experimental task conditions:

1. **Task 1:** Follow the path only
2. **Task 2:** Follow the path and avoid the obstacles
3. **Task 3:** Follow the path and collect the targets
4. **Task 4:** Follow, avoid, and collect together

There were no monetary rewards in the task. Since following paths, avoiding obstacles, and heading towards targets are frequent natural behaviors, we assume that subjects have some learned, and perhaps context-specific subjective values associated with the three task components, and our goal was to modulate these intrinsic values using the instructions. The instructions were to walk normally, but to give some priority to the particular task components in the different conditions. To encourage such prioritization, Subjects received auditory feedback when colliding with obstacles or targets. When objects were

task-relevant, this feedback was positive (a fanfare) or negative (a buzzer), while collisions to task-irrelevant objects resulted in a neutral sound (a soft bubble pop) [307]. The color of the targets and obstacles was counterbalanced in another version of the experiment and was found not to affect task performance or the distribution of eye fixations so the control was not repeated in the present experiment [128]. The order of the task was Task 1, 2, 3, and 4. This order was chosen so as not to influence the single-task conditions by doing the double task. Thus there may be some ordering effects. In another experiment in the environment, the order of the conditions was counterbalanced and no obvious order effects were observed [128].

We analyze data collected from 25 human subjects. A single experimental trial consisted of a subject traversing the room, with the trial ends when the goal at the end of the path is reached. Objects' positions and the path's shape differed on every trial. Each subject performed four trials for each task condition.

**Data availability** This general paradigm of navigation with targets and obstacles has been used to evaluate modular RL and IRL algorithms [289, 258] and to study human navigation and gaze behaviors [257, 307]. The data that support the findings of this study are made public and available at [306]<sup>3</sup>.

---

<sup>3</sup><https://doi.org/10.5281/zenodo.255882>

#### 4.4.2 Results

Despite its computational advantages shown in simulation, the question remains whether modular IRL can be used as a decision-making model to explain human behaviors in the experiments. Sparse modular IRL (Eq (4.9)) is used as the objective function to estimate reward  $r$  and discount factor  $\gamma$  for the target, obstacle, and path modules. However, the regularization constant is found to be close to zero since there are only three modules. Recall that each subject performs each task four times, and each time the path and the arrangement of objects are different. We use leave-one-out cross-evaluation, where  $r, \gamma$  are estimated using all-but-one training trials that are from the same subject and same task condition and evaluated on the remaining test trial. Since the parameter estimates are based on the other three trials, all of our prediction results shown below are for a *novel* environment with similar components – this requires the model to generalize across environments. The number of data samples obtained from a single trial is typically around 100 hence sample efficiency is critical for the performance of an algorithm.

Different  $r$  and  $\gamma$  are estimated for each subject under each task condition for each module, hence there are 25 subjects  $\times$  4 conditions  $\times$  3 modules  $\times$  4 trials = 1,200 different pairs of  $r, \gamma$  estimations. The state information for the model includes the distance and angle to the objects, while the state space is discretized using grids of size 0.572 by 0.572 meters, a parameter chosen empirically that produces the best modeling result. It also matches the approximate length of a step in VR, so is a suitable scale for human direction

decisions. Empirically, as long as the grid size is within a reasonable range of human stride length (0.3-0.9 meters) the algorithm’s performance is fairly robust.

The path is discretized into a sequence of waypoints that are removed after being visited (similar to the targets). The action space spans 360 degrees and is discretized to be 16 actions using bins of 22.5 degrees. This is a suitable discretization of the action space, given the size of the objects at the distance of 1-2 meters, where an action decision is most likely made.

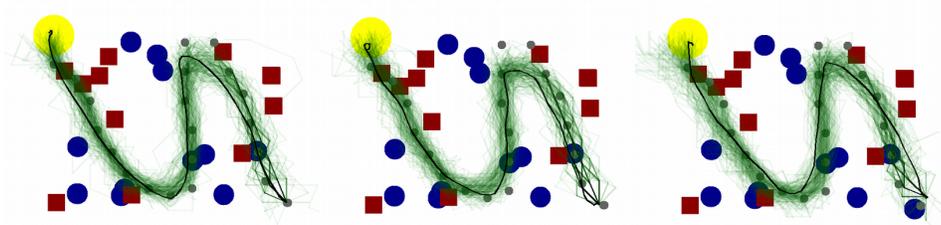
#### 4.4.2.1 Qualitative Results and Visualization

The most intuitive way to evaluate the modular RL model is to see whether the model can accurately reproduce human navigation trajectories. The Q-value function of a modular RL agent is calculated using  $r$  and  $\gamma$  estimated from human data. Next, the modular RL agent is placed at the same starting position as the human subject and starts to navigate the environment until it reaches the end of the path. The agent chooses an action probabilistically based on the Q-value of the current state, using a softmax action selection function as in Eq (4.4). The reason to let the agent choose actions with a certain degree of randomness is that the Q-values for multiple actions can be very close, e.g., turning left or turning right to avoid an obstacle, consequently a human subject may choose either. Therefore, a single greedy trajectory may not overlap with the actual human trajectory. The softmax action selection function generates a distribution of hypothetical trajectories, i.e., a trajectory

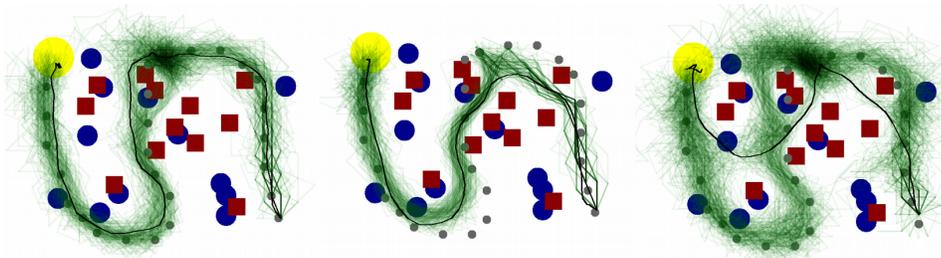
cloud, by running an agent many times in the same environment. The actual human trajectory can be visualized in the context of this distribution.

Fig 4.8 shows generated trajectory clouds together with actual human trajectories, along with estimated rewards and discount factors. The agent trajectories are shown in the semi-transparent green hence darker area represents trajectories with higher likelihood, and the human trajectory on that trial is shown in black. Each row of figures presents experimental trials from one experimental condition (Task 1-4), and three trials within each row are from different subjects but the same environment, i.e., the same arrangement of objects.

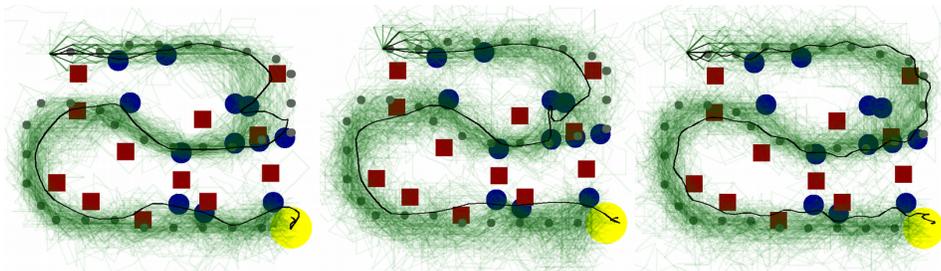
The figures demonstrate that the model’s generated trajectory clouds align well with observed human trajectories. When a local trajectory distribution is multi-modal, e.g., in Fig 4.8d, 4.8f, 4.8j, 4.8k, and 4.8l, the human trajectories align with one of the means. The next important observation is the between-subject variation. Trials within each row are from the same environment under the same task instruction. However, human trajectories can sometimes exhibit drastically different choices, e.g., Fig 4.8e versus 4.8f, 4.8j versus 4.8k. These differences are modeled by the underlying  $r$  and  $\gamma$  and accurately reproduced by the distributions generated. This means that we can compactly model naturalistic, diverse human navigation behaviors using only a reward and a discount factor per module. The modeling power of modular RL is demonstrated by the observation that varying these two variables can produce a rich class of human-like navigation trajectories.



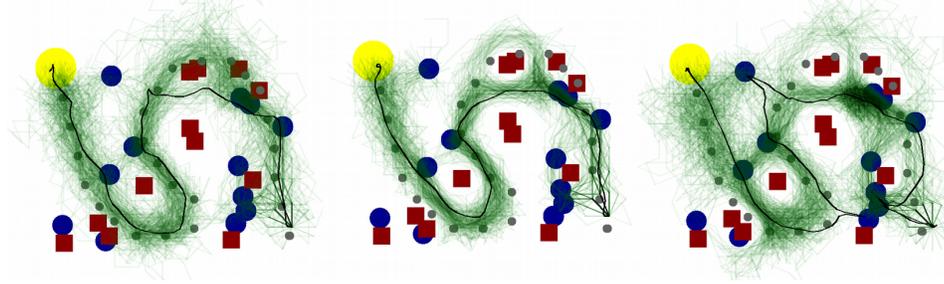
(a)  $r : (0.11, 0.38, \mathbf{0.52})$  (b)  $r : (0.11, 0.25, \mathbf{0.63})$  (c)  $r : (0.07, 0.39, \mathbf{0.54})$   
 $\gamma : (0.85, 0.99, 0.88)$      $\gamma : (0.88, 0.99, 0.94)$      $\gamma : (0.82, 0.99, 0.94)$



(d)  $r : (0.00, \mathbf{0.57}, \mathbf{0.43})$  (e)  $r : (0.02, \mathbf{0.57}, \mathbf{0.41})$  (f)  $r : (0.06, \mathbf{0.75}, \mathbf{0.19})$   
 $\gamma : (0.00, 0.69, 0.91)$      $\gamma : (0.99, 0.59, 0.97)$      $\gamma : (0.95, 0.60, 0.88)$



(g)  $r : (\mathbf{0.30}, 0.22, \mathbf{0.48})$  (h)  $r : (\mathbf{0.27}, 0.29, \mathbf{0.45})$  (i)  $r : (\mathbf{0.30}, 0.17, \mathbf{0.52})$   
 $\gamma : (0.77, 0.72, 0.89)$      $\gamma : (0.69, 0.73, 0.96)$      $\gamma : (0.76, 0.99, 0.89)$



(j)  $r : (\mathbf{0.28}, \mathbf{0.34}, \mathbf{0.39})$  (k)  $r : (\mathbf{0.12}, \mathbf{0.6}, \mathbf{0.28})$  (l)  $r : (\mathbf{0.12}, \mathbf{0.77}, \mathbf{0.11})$   
 $\gamma : (0.72, 0.76, 0.90)$      $\gamma : (0.74, 0.56, 0.97)$      $\gamma : (0.74, 0.53, 0.95)$

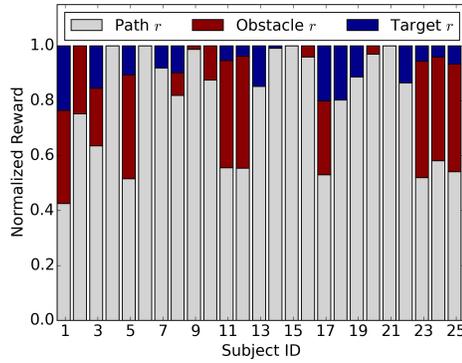
Figure 4.8: Bird’s-eye view of human trajectories and agent trajectory clouds across different subjects. Black lines: human trajectories. Green lines: modular RL agent trajectory clouds generated using softmax action selection. The green is semi-transparent hence darker area represents trajectories with higher likelihood. Yellow circles: end of the path. Blue circles: targets. Red squares: obstacles. Gray dots: path waypoints used by the model (subjects see a continuous path). Below each graph are the rewards and discount factors estimated from human and used by the modular RL agent. The rewards and discount factors are shown in the order of (Target, Obstacle, Path). The module rewards that correspond to task instructions are bold. Obstacle module has negative reward, but to compare with the other two modules the absolute value is taken. Three trials within each row are from different subjects but the same environment. (A,B,C) show trials from **Task 1: follow the path**. (D,E,F) show trials from **Task 2: follow the path and avoid obstacles**. (G,H,I) show trials from **Task 3: follow the path and collect targets**. (J,K,L) show trials from **Task 4: follow the path, collect targets, and avoid obstacles**.

Note that the human trajectories do not always fall at the exact center of the distribution, this is expected because the rewards and discount factors are estimated not from the shown single trial, but using multiple trials that belong to the same subject and the same task conditions to obtain enough data samples.

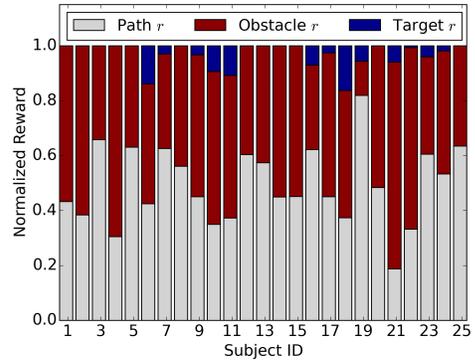
#### 4.4.2.2 Between-task and Between-subject Differences

We then look at the way average reward estimates vary between different tasks when aggregating data from all subjects. The results are shown in Fig 4.10a. Overall, the estimated  $r$  values vary appropriately with task instructions. Thus obstacles are valued higher when the instructions prioritize this task, and targets are valued higher when that task is prioritized. Note that the obstacle avoidance module is given some weight even when it is not explicitly prioritized – this is consistent with the observation that subjects deviate from the path to avoid obstacles even when obstacles are task-irrelevant. This may reflect a bias that is carried over from natural behavior with real obstacles. The relatively high value for the path may indicate that subjects see staying near the path as the primary goal.

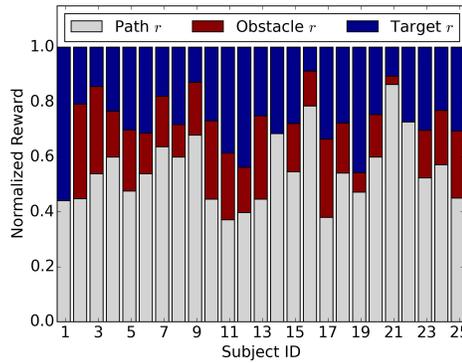
The between-subject differences in reward are shown in Fig 4.9 for all 25 subjects. At each individual subject’s level, changes in the relative reward between the modules are also consistent with task instructions. A one-way ANOVA test suggests that individual differences are evident across subjects under the same task instruction, shown in Table 4.3.



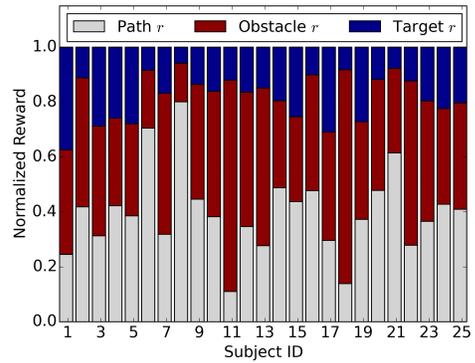
(a) Task1: path only



(b) Task2: obstacle + path



(c) Task3: target + path



(d) Task4: target + obstacle + path

Figure 4.9: Average normalized rewards for each subject under different task instructions. The relative reward magnitude changes between tasks and agrees with task instructions. Under the same task instruction, individual differences in reward function are shown.

	Target $r$	Obstacle $r$	Path $r$
Task 1	$F(25, 4) = 6.53$ $p = 3.38 \times 10^{-11}$	$F(25, 4) = 5.60$ $p = 1.16 \times 10^{-9}$	$F(25, 4) = 4.57$ $p = 8.44 \times 10^{-8}$
Task 2	$F(25, 4) = 8.09$ $p = 1.41 \times 10^{-13}$	$F(25, 4) = 12.11$ $p = 1.18 \times 10^{-18}$	$F(25, 4) = 12.12$ $p = 1.16 \times 10^{-18}$
Task 3	$F(25, 4) = 7.65$ $p = 6.11 \times 10^{-13}$	$F(25, 4) = 5.91$ $p = 3.50 \times 10^{-10}$	$F(25, 4) = 3.17$ $p = 4.50 \times 10^{-5}$
Task 4	$F(25, 4) = 21.38$ $p = 6.57 \times 10^{-27}$	$F(25, 4) = 5.03$ $p = 1.21 \times 10^{-8}$	$F(25, 4) = 7.20$ $p = 3.00 \times 10^{-12}$

Table 4.3: One-way ANOVA for individual differences in reward between subjects and across task instructions. Between-subject differences for all modules are significant in all task conditions.

Fig 4.10b shows average discount factor estimates for different tasks. Although the reward evidently reflects and agrees with task instructions, the interpretation of the discount factor is more complicated. The discount factors vary across tasks for target and obstacle modules but are close to 1.0 and stable for the path module. This may also reflect the primacy of the task of getting across the room, and the need to plan ahead. Although the instructions do not directly manipulate discount factors, we will later show that estimating discount factors from data instead of holding them fixed is important for modeling accuracy.

#### 4.4.2.3 Stability of Rewards and Discount Factors across Tasks

An important observation from Fig 4.10 is that *task-relevant* module rewards and discount factors are stable across task conditions. To show this quantitatively, for each subject, we combine module rewards from Task 2 (path

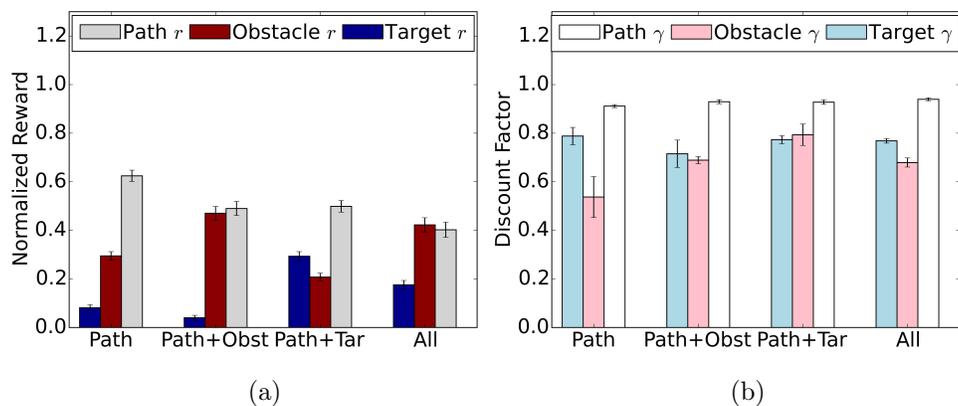


Figure 4.10: (a) Normalized average estimated rewards across different task instructions. The error bar represents the standard error of the mean between subjects ( $N = 25$ ). The obstacle module has negative reward, but to compare with the other two modules its absolute value is taken. The estimated reward agree with task instructions. (b) Average estimated discount factors across different task instructions. The error bar represents the standard error of the mean between subjects ( $N = 25$ ).

	Target $r$	Obstacle $r$	Path $r$
Task 2+3 synthesized	$0.177 \pm 0.018$	$0.415 \pm 0.028$	$0.408 \pm 0.021$
Task 4	$0.180 \pm 0.017$	$0.422 \pm 0.029$	$0.398 \pm 0.031$
	Target $\gamma$	Obstacle $\gamma$	Path $\gamma$
Task 2+3 synthesized	$0.773 \pm 0.017$	$0.689 \pm 0.015$	$0.928 \pm 0.006$
Task 4	$0.768 \pm 0.009$	$0.679 \pm 0.019$	$0.936 \pm 0.006$

Table 4.4: Task-relevant module rewards and discount factors are transferable across task conditions. The table shows synthesized rewards and discount factors compared to the estimated ones. Rewards are re-normalized. Results are presented as mean  $\pm$  standard error between subjects (N=25).

+ obstacle) and Task 3 (path + target) to synthesize the rewards for Task 4 (path + obstacle + target) in the following way:

$$r_{task4\_target} = r_{task3\_target} \quad (4.10)$$

$$r_{task4\_obstacle} = r_{task2\_obstacle} \quad (4.11)$$

$$r_{task4\_path} = (r_{task2\_path} + r_{task3\_path})/2 \quad (4.12)$$

Then the discount factors are synthesized similarly. The synthesized rewards (re-normalized) and discount factors from Task 2 and 3 are found to be very close to those estimated from Task 4, as shown in Table 4.4. However, task-irrelevant rewards and discount factors are not stable. This result indicates that task-relevant module rewards and discount factors generalize to a different task condition. Thus modules are independent and transferable in this particular scenario.

#### 4.4.2.4 Quantitative Results and Comparisons to Alternative Models

Next, we compare our model with several alternative hypotheses. The full modular IRL model chooses the action greedily that maximizes the Q-value function of each state using both estimated  $r$  and  $\gamma$ . An ablation study is conducted to demonstrate the relative importance of the variables in the model. The binary reward agent estimates  $\gamma$  only, and uses a unit reward of 1 for the module that is task-relevant, e.g., in Task 2 the path and the obstacle modules would have rewards of +1 and -1 respectively, and the target module would have a reward of 0. The fixed  $\gamma$  agents estimate  $r$  only, and use fixed  $\gamma = 0.1, 0.5, 0.99$ . A Bayesian IRL agent without modularization and assumes a fixed discount factor [244] is also implemented where the implementation details can be found in Appendix B.1. A Random agent serves as a baseline that chooses an action uniformly random.

We choose two performance metrics to evaluate these models. The first one is the number of objects intercepted by the agent’s entire trajectory under different task conditions. Fig 4.11 shows the performance of different models ((a) targets and (b) obstacles). Overall, the modular IRL model has the closest performance to the human data across task conditions. Note that the number of targets collected is only a little affected by the avoid instruction and obstacles avoided do not change very much with the target instruction, supporting the previous claim that the modules in this experiment are independent hence task-relevant module values are stable. Bayesian IRL and fixed  $\gamma = 0.99$  models

perform poorly—the number of objects hit does not vary accordingly with task instructions. The binary reward models,  $\gamma = 0.1, 0.5$  reflect task instructions correctly but are less accurate than the full modular IRL model.

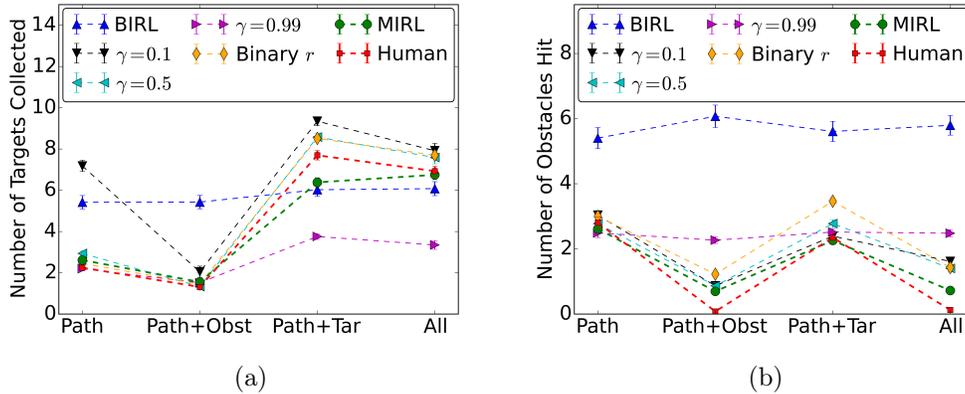


Figure 4.11: Average number of targets collected/obstacles hit when different models perform the navigation task across all trials. There are 12 targets/obstacles each in the virtual room. Error bars indicate standard error of the mean ( $N = 100$ ).

A quantitative evaluation metric would be the angular difference, i.e., policy agreement, which is obtained by placing an agent in the same state as a human and measuring the angular difference between the agent’s action and the human subject’s action. This metric differs from the previous one because it emphasizes more on the accuracy of local decisions instead of the whole trajectory. Thus this angular difference is a local metric instead of a holistic one. The comparison results are shown in Table 4.5. All modular RL agents are more accurate in predicting human actions comparing to the traditional Bayesian IRL algorithm. Again the full modular IRL model results in higher

	Task 1	Task 2	Task 3	Task 4
Random	90.15±0.66	89.03±0.59	89.49±0.61	90.72±0.59
Bayesian IRL	53.87±2.54	53.37±2.71	59.86±2.00	51.09±2.60
Fixed $\gamma = 0.1$	31.74±0.88	39.43±1.18	36.16±0.75	41.40±0.88
Fixed $\gamma = 0.5$	21.46±0.46	36.04±1.16	34.20±0.78	39.14±0.92
Fixed $\gamma = 0.99$	18.19±0.32	27.63±1.41	28.61±0.93	31.63±1.08
Binary Reward	17.66±0.38	27.66±1.44	29.97±0.72	29.80±0.95
MIRL (Full Model)	17.94±0.33	27.39±1.46	26.98±0.80	27.65±1.02

Table 4.5: Evaluation of the modular agent’s performance compared with baseline agents, measured by the average angular difference (in degrees) compared to actual human decisions. The results are presented as mean  $\pm$  standard error ( $N = 100$ ). The agent that uses the full model outperforms all other models.

accuracy comparing to the alternative models. The binary reward model has comparable performance and is in general better than the models that have the discount factor fixed. This supports our claim that module-specific discount factor plays an important role in modeling human behaviors and should be estimated from data.

To summarize, we can predict human novel trajectories in different environments based on rewards and discount factors estimated from behavioral data. Since we do not know the actual set of visual operations involved in walking through a cluttered room like this, the fact that we can reproduce the trajectories suggests that the three chosen modules can account for a substantial fraction of the behavior while vision may be used for other tasks. In fact, close to half the fixations made by the subject are on regions of the environment other than the path or objects [307]. This suggests that there may be other visual computations going on but that they do not have much

influence on the behavior. Thus the modular RL agents generate reasonable hypotheses about the underlying human decision-making mechanism.

These results provide strong support for using modular RL as the model for explaining such multitask navigation behaviors, and modular IRL as a sample efficient algorithm to estimate rewards and discount factors. Bayesian IRL has to deal with a complex high-dimensional state space and settle for its approximations for a dynamic multi-task problem with limited data, while modular RL can easily reduce the dimensionality of the state-space by factoring out sub-tasks. Therefore the algorithm significantly outperforms the previous standard IRL method in terms of the accuracy in reproducing human behaviors.

## 4.5 Discussion, Related Work, and Future Work

This chapter formalizes a modular reinforcement learning model for natural multitask behaviors. Modular RL is more suitable for modeling human behaviors in natural tasks while standard RL serves as a general model for reward-seeking behaviors. The two important variables in modular RL are module-specific reward and discount factor, which can be jointly estimated from behavioral data using the proposed modular IRL algorithm. A computer simulation demonstrated the validity and sample efficiency of the modular IRL. In a virtual-reality human navigation experiment, we showed multitask human navigation behaviors, across subjects and under different instructions, can be modeled and reproduced using modular RL.

#### 4.5.1 Relation with other Reinforcement Learning Models

The proposed modular IRL algorithm is an extension and refinement of [258] which introduced the first modular IRL and demonstrated its effectiveness using a simulated avatar. The navigation tasks are similar but we use data from actual human subjects. While they use a simulated human avatar and a straight path, our curved path proves quite different in practice, as well, being significantly more challenging for both humans and virtual agents. We then generalize the state space to let the agent consider multiple objects for each module, while the original work assumes the agent considers one nearest object of each module.

Bayesian IRL was first introduced by [244] as a principled way of approaching an ill-posed reward learning problem. Existing works using Bayesian IRL usually experiment in discretized grid worlds with no more than 1000 states with an exception being the work of [63] which was able to test on a goal-oriented MDP with 20,518 states using hierarchical Bayesian IRL.

The modular RL architecture proposed in this work is most similar to a recent work in [314], in which they decompose the reward function in the same way as the modular reinforcement learning. Their focus is not on modeling human behavior, but rather on using deep reinforcement learning to learn a separate value function for each subtask and combining them to obtain a good policy. Other examples of divide-and-conquer approach in RL include factored MDP [111] and co-articulation [253].

Hierarchical RL [79, 300] utilizes the idea of *temporal abstraction* to allow more efficient computation of the policy. [287] analyzes human decision data in spatial navigation tasks and the Tower of Hanoi; they suggest that human subjects learn to decompose tasks and construct action hierarchy in an optimal way. In contrast with that approach, modular RL assumes *parallel decomposition* of the task. The difference can be visualized in Fig 4.12. These two approaches are complementary, and are both important for understanding and reproducing natural behaviors. For example, a hierarchical RL agent could have multiple concurrent *options* [79, 300] executing at a given time for different behavioral objectives. Another possibility is to extend the modular RL to a two-level hierarchical system. Learned module policies are stored and a higher-level scheduler or arbitrator decides which modules to activate or deactivate given the current context and the protocol to synthesize module policies. An example of this type of architecture can be found in [289].

A similar approach to modular RL is factored MDPs [111]. The difference is that a factored MDP is a top-down, decomposition-based approach, in which the global MDP must be known beforehand. In contrast, modular RL is a bottom-up, ensemble-based approach. Another closely related approach is co-articulation [253], where a precedence relationship between components is used to determine a global policy.

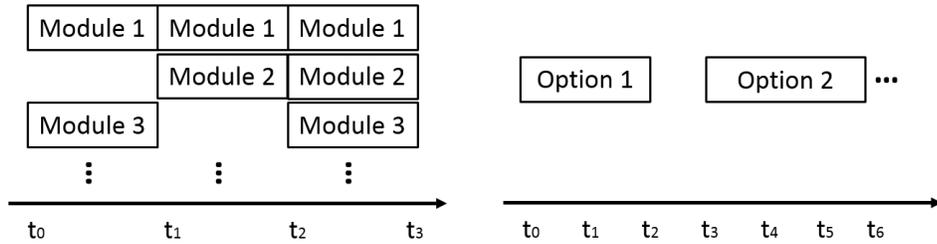


Figure 4.12: Modular reinforcement learning (left) vs. hierarchical reinforcement learning (right). Modular RL assumes modules run concurrently and do not extend over multiple time steps. Hierarchical RL assumes that a single option may extend over multiple time steps.

#### 4.5.2 Implications

Modular RL/IRL makes it possible to estimate the subjective value of particular human behavioral goals. Over the last 15 years, it has become clear that the brain’s internal reward circuitry can provide a mechanism for the role of tasks on both gaze behavior and action choices. It is thought that the ventromedial prefrontal cortex and basal ganglia circuits encode the subjective values driving behavior [184, 36, 349]. The present work shows that it is possible to get a realistic estimate of the subjective value of goals in naturalistic behavior, and these values might reflect the underlying reward machinery. Many of the reward effects observed for neurons have very simple choice response paradigms. Thus it is important to attempt to link the primary rewards used in experimental paradigms and the secondary rewards that operate in natural behavior. Previous human experiments have typically used simple behaviors with money or points as rewards. In our experiment, we used instructions to bias particular aspects of basic natural behavior with no explicit rewards.

The results provide support for a modular cognitive architecture when modeling natural visually guided behaviors. Modularization reduces the size of state space and alleviates the curse of dimensionality. Consequently, modular IRL is more sample efficient than the standard Bayesian IRL. In addition, modular RL estimates a discount factor for every module hence it is more flexible and powerful than a standard RL model in which the discount factor is unitary and fixed. The modeling result suggests having such flexibility is indeed helpful. It may also explain why basal ganglia have the mechanism to implement multiple discount factors [301].

The decomposition of the global task also allows humans to reuse a learned module later in a new environment. This claim is supported by the observation that task-relevant module rewards and discount factors are stable and generalize to a different task condition. When immersed in a new environment, the simple form of Eq (4.3) allows the value function to be computed with a reasonable cognitive load. Subjects may learn stable values for the costs of particular actions like walking and obstacle avoidance and these subjective values factor into momentary action decisions [118]. For example, humans direct gaze to nearby pedestrians in a simple uninstructed walking context with a probability close to 0.5, with small variability between subjects [158] and a similar gaze distribution was found in a virtual environment [157]. These values may change in more complex contexts, as in the decoy effect for example [137]. The present work provides a way of testing the circumstances in which such subjective values might change.

Modular RL allows intuitive interpretation for multitasking behaviors, where relative importance and reward discounting rates can be compared between modules directly. We expect this modular approach of RL can be applied to and can explain many natural tasks. [22] has shown that a wide range of human behaviors can be modeled as consisting of microbehaviors, so many behaviors are a mixture of simple modules and could potentially be modeled in this way.

### 4.5.3 Limitations of the Model and Future Work

Although modular RL/IRL can produce trajectories that are similar to human behavior, the match was imperfect as demonstrated by the angular difference. One difficulty with modeling human behavior is that we defined the state space and a set of modules by hand without knowing the actual state representation or task decomposition that the human uses. This may account for the discrepancy between the human and agent policies. Ideally, we could learn state representation from data, but this involves the challenging task of combining representation learning and IRL. The work in [18] provides a potential method for inferencing goals and states for the modules. The recent development in deep reinforcement learning [209] may possibly lead to a data-driven approach to IRL that can learn state representation from data.

Another concern is whether the modules are indeed independent. For example, a subject might choose to ignore a target if it is close to an obstacle that might be difficult to avoid. Similarly, energetic costs may affect all

modules. For example, cutting a corner that has a target close by might be particularly appealing. In a similar experimental context, [307] observed deviations from strict independence between modules. Thus we should consider the modular approach as a first approximation that may help provide insights into naturalistic sensory-motor behavior.

An important consequence of being able to get a quantitatively estimated subjective reward and discount factor of a module is that it is possible to test whether these values are stable across contexts. For example, the value of avoiding an obstacle should be stable across moderate variations in the environment such as the changes in obstacle density or changes in the visual appearance of the environment. If this is true, then it is possible to make predictions about behavior in other contexts using learned modules. And it would also be possible to use the prediction error to indicate that other factors need to be considered.

## 4.6 Conclusion

How does the brain learn and make decisions to achieve behavioral goals in an information-rich environment, with limited cognitive resources?

It is generally agreed that human actions can be formalized within the framework of statistical decision theory, which specifies a cost function for action choices, and that the intrinsic value of actions is controlled by the brain's

dopaminergic reward machinery. Standard reinforcement learning methods were developed for artificial intelligence agents, and incur too much computation to be a viable model for real-time human decision making. We propose an approach called modular reinforcement learning that decomposes a complex task into independent decision modules. This model includes a frequently overlooked variable called the discount factor. We develop an algorithm called modular inverse reinforcement learning that estimates both the reward and the discount factor. We show that modular reinforcement learning may be a useful model for natural navigation behaviors. The estimated rewards and discount factors explain human walking direction decisions in a virtual-reality environment. This framework provides a potentially useful tool for exploring the task structure of natural behavior, and investigating how momentary decisions are modulated by internal rewards and discount factors.

How can we improve current artificial intelligence (AI) by studying these mechanisms of the brain, so that AIs can cope with the complexity in the real world?

Modular RL is not only a decision-making model of humans but also a candidate algorithm for RL research. We have seen that complex human navigation trajectories in novel environments can be reproduced by an artificial agent that is based on the modular model. In simulation, we have shown the advantage of modular RL/IRL over traditional methods in complex task domains with high-dimensional state space.

The computational model of humans also allows artificial agents to model and predict human behaviors better, which is useful in human-AI interaction tasks. Follow-up work with colleagues [159, 365] uses the modular RL model as the pedestrian’s decision model to predict their walking directions. With this information, simulated robots can predict human movements, synthesize a provable safe navigation policy under safety constraints, and navigate safely around pedestrians.

## Chapter 5

### Neural Basis of Modularization and Attention<sup>1</sup>

In previous chapters, we have discussed the modeling effort to understanding modular attention mechanisms at abstract levels. At a high level, modularization implies multiple, coexisting neural processes, and attentional control implies a neural mechanism that manages resources (neurons) for these processes. It is natural to ask how modules, or visual attention, are implemented in a biological cortical network with a large number of neurons. In this section, we discuss the neural basis of modularization and attention. The analysis here corresponds to level I of Marr’s paradigm:

How can the representation and algorithm be realized physically  
[199]?

A complete answer to this question is certainly not feasible in this work, hence we will try to address the implementation from a neuronal communication perspective.

---

<sup>1</sup>This chapter of work is based on the following publication: Ruohan Zhang and Dana H Ballard. Parallel neural multiprocessing with gamma frequency latencies. *Neural Computation*, 32(9):1635–1663, 2020. The dissertator is the first author, and takes the leading role in conceiving and designing the analysis, contributing analysis tools, performing the analysis, and writing the paper.

A fundamental problem here is that we only have a partial understanding of the basic communication protocols that underlie signal transmission. This makes it difficult to interpret the significance of particular phenomena such as basic spike firing patterns and oscillations at different frequencies. There are, of course, useful models. Poisson statistics of cortical action potentials have long been a basic component in models of signal representation in the cortex. The Poisson variability in cortical neural responses has been typically modeled using spike averaging techniques, such as trial averaging and rate coding since such methods can produce very reliable correlates of behavior. However, mechanisms that rely on counting spikes could be slow and inefficient thus they might not be useful in the brain for computations at timescales in the ten-millisecond range.

This issue has motivated a search for alternative spike codes that take advantage of spike timing and has resulted in many studies that use synchronized neural networks for communication. Here we focus on recent studies that suggest that the gamma frequency may provide a reference that allows local spike phase representations that could result in much faster information transmission. We have developed a unified model (gamma spike multiplexing, or GSM) that takes advantage of a single cycle of a cell's somatic gamma frequency to modulate the generation of its action potentials. In particular, this method of coding allows multiple independent processes to run in parallel, thereby greatly increasing the processing capability of the cortex. Intuitively, the theory hypothesizes that neurons communicate through 30-80 hertz gamma frequency

in a manner similar to radio stations. A group of neurons designated to a particular computational process could tune to a particular frequency in that range. It is therefore possible to form multiple separate networks that might constitute neural trains of thought that can be kept from crosstalk. This model suggests that the attention mechanism may reflect the use of additional neurons by a computational process by switching these neurons to the frequency band associated with that process. System-level simulations and mouse cortical cell data have shown that some prerequisites for the proposed theoretical model are met. If this theory is borne out by additional experiments, it would constitute a significant advance in both neural coding and artificial neural networks.

## 5.1 Summary of Contributions

1. We refine a phase coding model that allows a single spike to represent an analog quantity (Section 5.3.1).
2. We propose a parallel probabilistic neuron selection algorithm that allows multiple neural processes to recruit neurons for their individual computations in parallel (5.3.2).
3. We propose a frequency-band communication protocol that allows multiple neural processes to coexist without crosstalk in a single network (5.3.3).
4. We draw preliminary evidence from computer simulations (5.4) and mouse neural recording data (5.5) to support the above models.

## 5.2 Background and Motivation

The earliest action potential recordings played a crucial role in characterizing the receptive fields in the striate cortex, but with the study of whole circuits [252, 65] the role of oscillations in modulating action potentials has become more important. The possible role of timing-based code was due to [5]. Subsequent studies showed large areas of synchronization transiting cortical maps [251, 284, 271], which has been refined into a general view of coherent communication [331, 93]. Now we can study a pool of synchronized cells representing a specific computation that we will denote as a neural *process*.

With this perspective, we can ask the question: Is it possible to have multiple cortical processes in parallel? This question surfaced twenty years ago known as the binding problem: If, in the cortex, “red” neurons and “blue” neurons are active along with “square” neurons and “circle” neurons, how does the cortex distinguish the one color-shape pairing from the other? At the time the great majority recognized the need to solve this problem, but no solutions were proposed [254]. The central problem is partitioning the extensively connected circuitry in a way that avoids possible cross-talk between processes that need to be kept separate.

One easy answer would be to only allow one process to be active at a time. However, this solution is unlikely given the myriad of different computations in brain circuitry. Additionally, there is significant evidence for parallel processes. For example, in modeling the multiple spike recordings in a monkey tactile discrimination experiment, [167] developed a novel analysis

technique that can isolate the spikes that account for the variance in task parameters. Their analysis shows that only 14% of the spikes can account for all the task variance. However, from the perspective of multiprocessing, the remaining 86% of the spikes are presumably doing something else than the controlled task. A second example is Cisek’s monkey experiment in motor choices [66]. A monkey knows that they will have to choose one of the two movements, but must keep both simultaneously active until the appropriate cue. Neural recordings show separated activities for the possible choices [66]. To distinguish between them, presumably, there must be a mechanism to prohibit possible interference between the two.

To allow a cortical network to perform multiprocessing, we propose to use *multiplexing*, wherein the meaning of a spike generated by a neuron can change on short timescales, i.e., several active processes may time-share this neuron. We propose a neural coding model called gamma spike multiplexing (GSM) that adopts an efficient phase coding strategy at the level of single cells. We further show that neural networks can be set up in a very novel way that allows several processes to run in parallel. Our multiplexing model is unique by using the following conventions in combination:

1. Each process uses a unique gamma frequency;
2. Cells in a process are chosen probabilistically on each gamma cycle;
3. Cells can participate in more than one process at different cycles;

4. The state of a process is represented by different cells on each gamma cycle.

A key element of the GSM model is a phase coding model using gamma oscillations in the somatic membrane potentials as references. The local phase coding model was suggested over twenty years ago [131]. The proposed phase coding model resembles several existing models [317, 319, 217, 19] which allows each spike to represent an analog quantity.

To validate the proposed GSM model, we first demonstrate in a simulation that this multiplexing model allows separate processes to be executed in parallel successfully. The mixing process can still result in individual neurons exhibiting classical Poisson statistics. We then analyze mouse cortical cell data recorded with the patch-clamp technique and show that several preconditions of the GSM models are biologically plausible. The patch-clamp technique, e.g., [236], allows access to the fine structure of a cell's membrane potential in awake animals. This capability is necessary for studying gamma oscillations within the membrane potential, which can have very small amplitudes of a few millivolts and can be challenging to separate from large-scale network modulatory oscillation signals [93, 95, 59]. Advances in patch-clamp membrane temporal resolution have allowed us to extract the local spike phase automatically from intracellular data [236] and provide evidence for our model.

## 5.3 Gamma Spike Multiplexing Model

The GSM model has three major components. At the single-cell level, we refine a phase coding model that uses gamma frequency oscillation as a reference. At the network level, we describe how neurons are selected by multiple neural processes in parallel. We then propose a novel multiplexing model that allows a group of neurons to temporarily set up a private network with a designated frequency.

### 5.3.1 Gamma Frequency Phase Coding Model

Various population coding models that assume rate coding [98] have shown enormous usefulness in producing correlates of behavior. Meanwhile, positing the use of population coding as the cortex’s generative model is not without debate. Any explicit aggregation of spikes requires counting significant numbers of spikes that is expensive both in spikes and synapses [180]. A simple theoretical analysis in Appendix C.1 shows that, with a basic rate coding model, it may take a large number of neurons or spikes to signal a scalar value reliably.

Alternatively, we discuss a gamma frequency phase coding model [131, 217, 19]. The advantage of this model is that it allows a single spike to represent an analog quantity instead of a digital quantity (0 or 1), thereby making our model more efficient. The concept is shown in Fig. 5.1b. The spike’s delay code  $\Delta$  is computed as the interval between the spike and the beginning of a gamma cycle. The closer the spike to the start of the cycle, the larger the

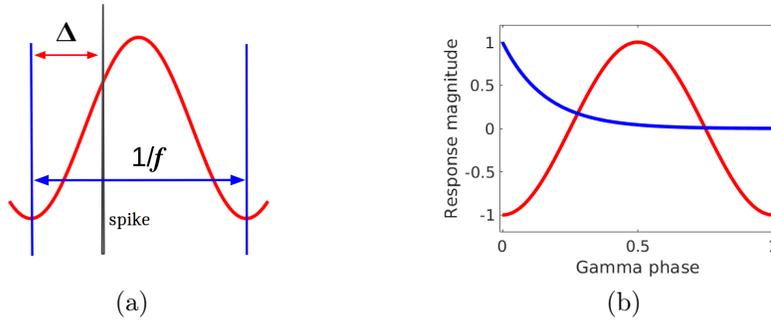


Figure 5.1: (a) Gamma frequency phase coding model. The red curve indicates the somatic oscillation in the range of gamma frequency. The spike’s latency code  $\Delta$  is then computed as the intervals between the gamma trough and the spike. (b) A spike signals an analog number (blue) coded as a phase delay from the trough. Short delays represent large magnitudes.

quantity it represents. The magnitude of response  $r$  is computed by

$$r = \exp(-\alpha l) \tag{5.1}$$

where  $l$  is the gamma phase and  $\alpha$  is a constant, as shown in Fig. 5.1a.

The ability of a phase coding model to represent an analog quantity using a single spike makes it significantly more efficient than the basic rate/population coding. However, an unresolved issue is to select the gamma cycle starting point. Guided by the phase precession model for place cells [224, 285], we use the gamma trough as shown in Fig. 5.1a. This choice is further resolved by analyzing experimental data. The detailed analysis with mouse cortical cell data that supports this claim is provided in Section 5.5. This model will serve as the basis for the rest of our multiplexing model.

### 5.3.2 Probabilistic Parallel Neuron Selection

Given that a spike can represent an analog quantity, we then discuss how the cortex selects neurons to participate in computational processes. The GSM model assumes that the cortical maps are over-complete and neurons are selected probabilistically in parallel.

#### 5.3.2.1 Sparse Coding Strategy

The first assumption is the over-completeness of cortical maps, that is, the cortex contains many times the number of neurons that would be mathematically needed to code an input. Consequently, a sparse coding strategy asserts that the early visual cortex develops neurons with receptive fields to represent the statistics of natural images, and, when coding a stimulus, relatively few neurons are chosen to be active at any one time [225, 226]. The over-completeness property reduces the probability of competing for the same neuron by multiple processes.

#### 5.3.2.2 Probabilistic Selection

The neurons are selected to code stimuli probabilistically with odds related to their receptive field projections. Given a target stimulus (represented as a vector), a neuron's response magnitude  $r_i$  can be calculated as the projection of the basis function it represents<sup>2</sup> onto the stimulus vector. A standard

---

<sup>2</sup>A neuron's vector of synaptic strengths is referred to as a basis function.

choice is to use a Boltzmann distribution where a cell is chosen with probability

$$p(r_i) = \frac{\exp(-\eta r_i)}{\sum_j \exp(-\eta r_j)} \quad (5.2)$$

where  $\eta$  is its ‘temperature’ parameter. The relative magnitudes of the input’s projections determine the probabilities of a neuron being selected to be part of the representation. The divisive normalization can be done in one step with lateral connections between neurons. Selecting neurons in this way ensures that all neurons participate in the receptive field learning process appropriately [152]. If the cell with the highest response is always selected as in maximum likelihood selection, unselected cells are never able to adjust their receptive fields.

The probabilistic selection has an even more significant consequence. If the stimulus is to be maintained for other gamma cycles, its code must be chosen anew on each such cycle. The state of a process migrates to a new group of cells on each iteration.

### 5.3.2.3 Parallel Selection

Given that a single candidate neuron is chosen probabilistically, one still needs to find a set of neurons to represent a stimulus. The standard residual-based method computes this set sequentially. However, in the GSM model, each step of sequential computation would require one gamma cycle, thus ruling it out as practical. To overcome this problem, we propose a parallel selection algorithm. Figures 5.2a and 5.2b compare the sequential and parallel

approaches. The stimulus vector (black) and basis functions of each neuron (gray) are represented as 2D vectors. Figure 5.2a shows the sequential method. The contribution of a cell (red) selected for representing the image can be determined by subtraction (green). In contrast, the parallel method Fig. 5.2b chooses a fixed number of cells probabilistically using Eq. 5.2 in parallel. In Appendix C.2, we assess the accuracy of the probabilistic parallel selection algorithm in simulation, but for the moment an important consequence of the parallel selection is that such representation is non-stationary, as shown in a related context by [81].



Figure 5.2: Parallel coding model. (a) A standard algorithm for coding an input (black arrow) picks a neuron (red) and sequentially fits the residual (green), a process that requires multiple iterations. (b) The GSM chooses several representing neurons in a single parallel step.

### 5.3.3 Neural Multiplexing Model

Once the phase coding and neuron selection mechanisms are in place, a very elegant way of instantaneously connecting large networks implicitly is possible, as shown in Fig. 5.3. If a selected subset of cells were co-modulated with the same gamma frequency simultaneously, then all the neurons in that

cohort would be behaving as engaged in the same computation. An analogy that helps understand this model is radio communication. A group of neurons can tune to a particular frequency to establish a private communication channel, although gamma oscillations are not used as a signal carrier but as a temporal reference. Collisions are handled by a gating mechanism depicted in Fig. 5.3b. If a neuron is chosen by two or more frequencies, it will not be allowed to spike. Although the exact mechanism is unknown, it is conjectured that such a mechanism is biologically plausible, for instance, through an interneuron's synaptic gating function [161, 121, 84, 145, 100], which might use sub-threshold signals [38].

A prerequisite for the model being able to support multiple computations is the ability of a neuron to modulate its oscillation frequency. It has already been observed that the gamma signal can change in a single cycle *in vivo* [13, 236]. While there is still no complete understanding of how the neural circuitry could control this functionality, [13] have suggested that the needed gamma modulation can be controlled by adjusting excitation and inhibition. Therefore the membrane potential can be modulated instantaneously. This demonstration is complemented by the circuit model of [45] that illustrates these features and the additional possibility of  $\sim 10$  ms spike phase delays. Additionally, another possibility is that some interneurons may contribute to columnar synchronization of activity in the gamma frequency range [192]. In Section 5.5, we provide experimental evidence that frequency modulation is indeed possible.

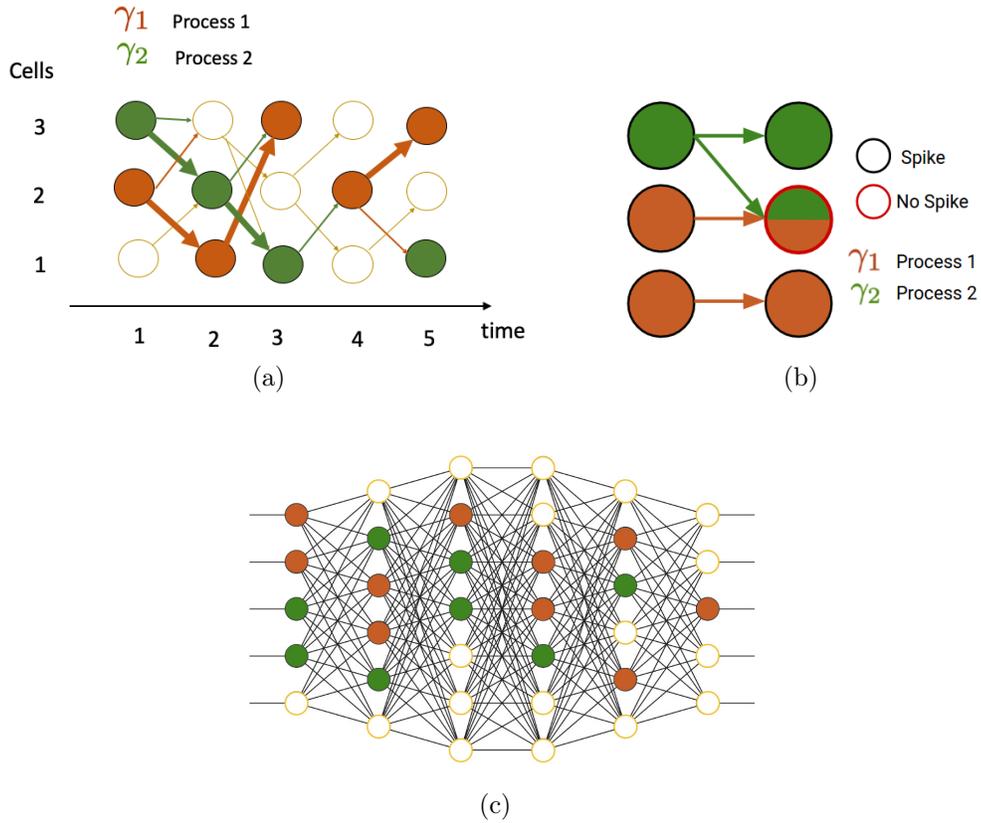


Figure 5.3: Neural multiplexing model. (a) The figure shows 3 neurons each depicted at five successive time instants. The somas are being modulated by two particular gamma frequencies at different times. Consequentially those neurons that are connected at the same frequency can communicate at those instants privately. A neuron can be chosen by different frequencies at different times. (b) Two timesteps in neural processing with two processes. If one neuron is chosen by two gamma frequencies, it does not spike. Thus only neurons that are chosen by a single gamma frequency can spike. If cells are not chosen by a gamma frequency, they do not spike. (c) At any given time, the network is effectively partitioned into multiple separate neural processes, within each the neurons communicate using their private gamma frequency band.

## 5.4 Simulation Results: A Sparse Coding Example

In a general multiplexing situation, one would expect multiple routines to be instantiated at the most abstract forebrain levels and migrate quickly throughout cortical maps, ultimately polling the lateral geniculate nucleus (LGN) for visual input. To sidestep the enormous effort it would take to build a model at this level of completeness, we use a much simpler model—the familiar computation of representing small image patches with receptive fields in the striate cortex using sparse coding [225, 226]. Figure 5.4 provides concrete examples of the neural process we simulated, i.e., coding an image patch with 50 neurons. The “LGN” generates the image patches that are sent to the simulated visual area V1 and encoded. Each “V1” neural process selects a subset of learned basis functions to encode a single image patch. The “V1” basis functions are learned using the algorithm by [179]. Appendix C.2 includes more examples of learned basis functions and coded image patches.

The basic setting of multiplexing is thus to code multiple image patches in a short time window. At any given discrete timestep, several image patches need to be coded simultaneously.

First, we need to calculate each neuron’s response for an image patch and convert the response to a delay code. We use  $r_i$  to denote the  $i$ th neuron’s response to its input,  $\mathbf{x}_i$  to denote the input vector, and  $\mathbf{w}_i$  to denote the recipient neuron’s basis function. A conventional way to calculate  $r_i$  is  $r_i = g(\mathbf{x}_i \cdot \mathbf{w}_i)$ , where  $g$  is a nonlinear activation function and only positive responses are maintained. The response is converted to a delay as shown in Fig. 5.1a.

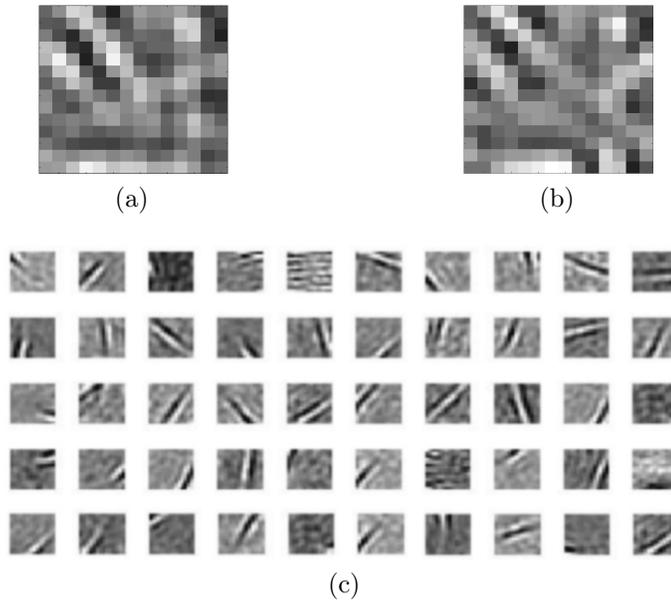


Figure 5.4: A sparse coding example. (a) In the model, 196 LGN cells appropriately filtered provide input for a small patch sampled from a camera. Their analog values can be depicted abstractly as a vector as shown by the black arrow in Fig. 5.2b. (b) The LGN input is used to choose cortical cells by randomly sampling cells' probability density function of the input projections. (c) 50 neurons (basis functions) selected by the parallel probabilistic algorithm to code the image patch. The basis functions are learned using the sparse coding algorithm [179].

Coding neurons for a particular patch are selected using the aforementioned probabilistic parallel selection algorithm.

Table 5.1 shows the parameters chosen for the simulation. It is important to choose parameters that are biologically plausible for our later analyses:

1. The total number of processes  $N_p$ . In primates, the number of simultaneously active processes may be linked to working memory, and in that case, the number would be four [195]. However much less is known about the use of unconscious, over-learned memory, which may have a significantly higher figure which we choose to be 16.
2. The number of gamma cycles used by an individual process  $t_\gamma$ . This parameter is reasonably guided by the duration of gaze fixation which indicates the length of a visual process, typically having a mean of  $200 \sim 600$  milliseconds.
3. The time needed for the delay code  $d_\gamma$ . This choice is bounded by the observation that sensitivity to the Pulfrich pendulum illusion [190] that implies a sensitivity to  $5 \sim 20$  milliseconds and the need to keep the phase delay less than half the gamma cycle.

In the simulation, the initial step is to specify the number of processes. For each process, a start time, dedicated gamma frequency, and duration are chosen. Next, its coding neurons are selected. If a neuron is chosen for a process, during the time for the delay and following refractory period it is marked as in use and cannot be selected by another process.

Name	Value	Unit	Description	Function
$T$	800	milliseconds	total length of simulation	simulation
$n_b$	1000	scalar	total number of neurons	simulation
$n_s$	$14 \times 14$	pixels	dimension of input and basis functions	simulation
$n_r$	50	scalar	number of neurons to code the stimulus	simulation
$N_p$	16	scalar	number of processes	process
$f_p$	$40 + \mathcal{N}(0, 7)$	Hz	frequency to modulate a process	process
$t_\gamma$	$\text{rand}(20, 22)$	cycles	process duration	process
$d_\gamma$	5	milliseconds	max length of the delay	spike
$d_r$	4	milliseconds	refractory period	spike

Table 5.1: Parameter values used in the simulations. The function  $\text{rand}(a,b)$  returns a single uniformly distributed random number in the interval  $[x,y]$ .  $\mathcal{N}(\mu, \sigma)$  is the normal distribution.

#### 5.4.1 Validating the Coding Algorithm

We first validate our implementation of the sparse coding algorithm [179], and more importantly, the parallel probabilistic selection algorithm. Appendix C.2 provides a concrete example of the neural process we simulated, i.e., coding several image patches in parallel with 50 neurons for each patch. It further shows that the parallel probabilistic selection algorithm results in good coding quality with enough neurons (50 or more). With 50 neurons, the mean coding error per pixel is  $4.0 \times 10^{-3}$  (standard error =  $0.4 \times 10^{-3}$ , which is considered small given pixel values are in the range of  $[0, 1]$ ).

#### 5.4.2 Poisson Statistics and Parameter Sensitivity

We now present the main simulation results. The most basic firing pattern of cortical cells exhibits Poisson distributions [97, 273, 34]. These distributions comprise the core of various coding models. Therefore, any proposed theoretical model must obey Poisson statistics in simulation, regardless

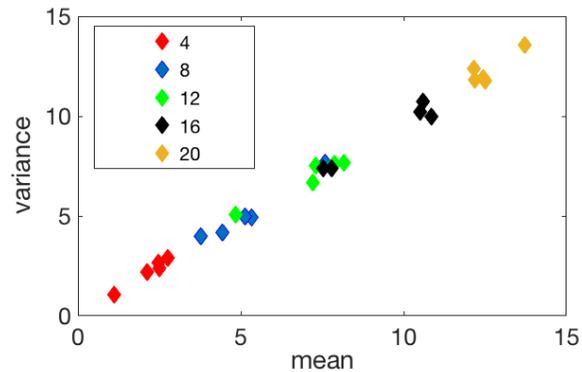


Figure 5.5: The GSM model generates spikes that appear Poisson distributed in simulation. We calculate the Fano ratio (the ratio between the variance and the mean of a random process) of simulated spike train data while varying the number of processes in simulation. The number of processes varied from 4, the putative value for short-term memory, up to a maximum of 20. Five samples were measured for each setting. The Fano ratio for the simulation samples are very close to an ideal Poisson process which has a Fano ratio of 1.

of the parameters chosen. A standard way to test this is to calculate the Fano ratio, i.e., the ratio between the variance and the mean of a random process in some time window [71]. For a Poisson process, the variance in the count equals the mean count, so its Fano ratio is 1.

The important parameter here is the number of processes. In terms of generating spikes that appear Poisson distributed, the GSM model is surprisingly insensitive to the parameter choice, as shown in Fig. 5.5. It is important to emphasize that many sets of parameters would work to illustrate that the GSM coding strategy produces distributions of spikes that appear Poisson on long timescales.

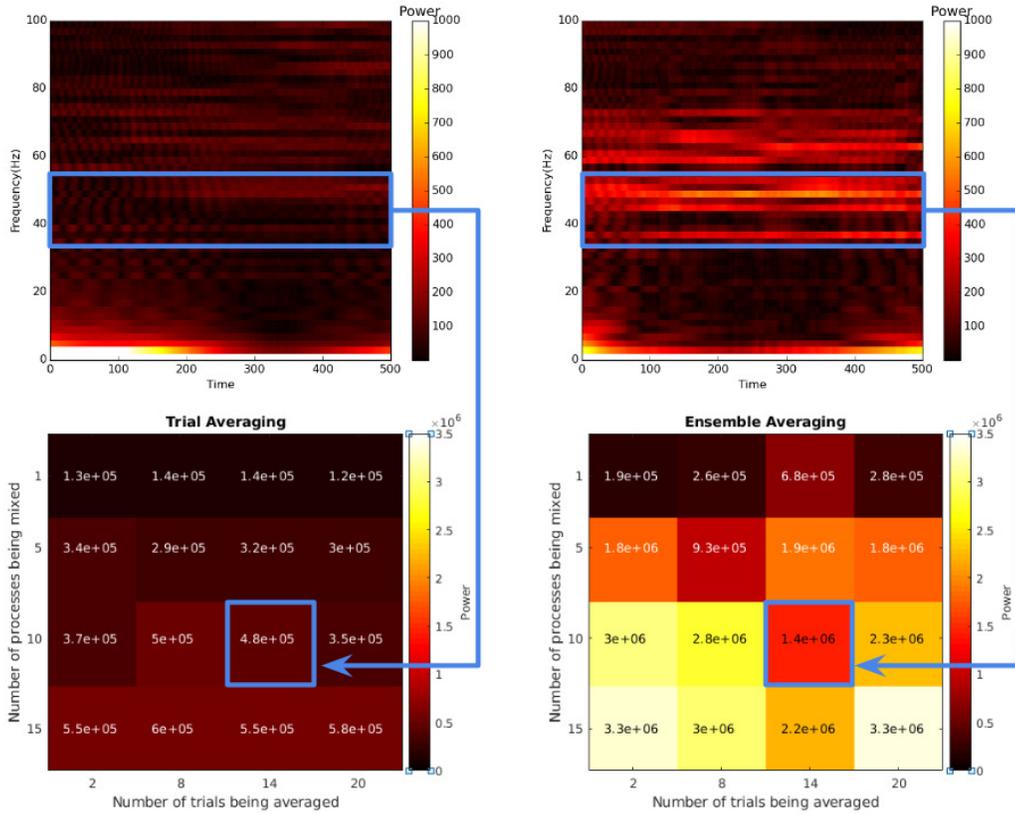


Figure 5.6: Gamma oscillation is less visible when performing trial averaging. Top row: Fourier spectrograms for the trial averaging vs. the ensemble averaging. Top left: Spectrogram for the trial averaging cases reveals very little power in the gamma band because the controlling parameters are different for each trial. Top right: The ensemble averaging spectrogram reveals marked frequency responses in the gamma band as each process is using a common set of parameters. Bottom row: The effect of the number of processes and number of trials on the aggregated power. The power is aggregated between the 35-55Hz gamma frequency band and over time. The key observation here is that gamma oscillation in the ensemble averaging case is consistently more visible than the trial averaging case regardless of parameters chosen, especially when we mix more than one process. This simulation explains why in experiments gamma oscillation is sometimes difficult to detect when performing trial averaging.

### 5.4.3 Observability of Gamma Oscillation

The proposed model, if true, would drastically change the way we interpret spike data. Now individual spikes can be identified with the specific image patch that being coded and can be readily associated with their individual processes. Appendix C.3 presents a visual comparison between the old and the new ways of interpreting spike train data. This new interpretation could potentially resolve a highly debated issue that gamma frequencies are observed in some experiments but not others. This phenomenon has lead researchers to doubt the role of gamma oscillation [246] or even raise the question of whether it is in fact neural noise [47].

In the classical view, data is acquired by trial averaging of a single neuron where the histories from different trials are summed once a common temporal reference is determined. The main assumption is that the overall process is *ergodic*, that is, trial averaging is deemed to be equivalent to averaging many cells in a large network, but according to the GSM model this equivalence does not hold. From the GSM perspective, trial averaging will have different controlling parameters (those in Table 5.1) from trial to trial. In contrast, data could also be acquired by ensemble averaging which uses the same control parameters for all the participating cells from a single trial. The result is that in ensemble averaging, correlations between neurons' oscillations over time can be evident, whereas in trial averaging, they can be weak or nonexistent.

The GSM model allows the evaluation of the extent to which gamma frequencies are observable using simulations. Figure 5.6 shows the Fourier

spectrogram of the trial average case. As the simulation shows, the gamma power is all but absent from the spectrogram. In contrast, Fig. 5.6 further shows the case where the averages are taken in an ensemble, where all coding cells use the same control parameters. The gamma power is very evident. The GSM model obviates the assumption that the neural population in an ensemble obeys ergodicity where repeatedly sampling one cell is equivalent to sampling a population simultaneously. The bottom line of this spectrogram analysis is to point out that the methodological differences between ensemble averaging and trial averaging may be the reasons why gamma signals are observed in some experiments but not others.

## 5.5 Evidence from Neural Recording Data

We have shown that the GSM model agrees with experimental observations in simulation. The full model can not be tested until recording technology allows us to perform intracellular recordings of multiple nearby neurons of an awake animal. However, we can test whether some prerequisites of the model are met.

The effects of sub-threshold membrane potential can vary [64], but the modulations of spikes by gamma oscillations have been observed at least almost thirty years ago [192], where the spike timings were modulated by the gamma oscillations. The recording techniques have greatly improved since then. We used a dataset obtained by the Gentet laboratory [236] of two-photon targeted patch-clamp recordings in area V1 layers 2/3 of awake mice at 20kHz. We

analyzed the recorded somatic membrane potential of 9 pyramidal cells with 536 trials in total. Gamma frequency oscillations (30-80Hz) are extracted by a Butterworth bandpass filter after spikes are removed. The spike-removal technique was validated experimentally to make sure that it does not produce undesirable artifacts when estimating the spike phase. Appendix C.4 shows a representative trial, including the spikes, the somatic potentials, extracted gamma oscillations, and the spike phases.

When we introduced the gamma phase coding model, there was an unsettled issue of picking the gamma cycle starting point. An important observation from the data is that 88.7% of the spikes are between the trough and the peak of a gamma cycle. By selecting the trough as the cycle starting point, the majority of the spikes are in the first half of a gamma cycle, so the delay is in the range of approximately 6 ~ 16ms (assuming 30-80Hz) as required by the model. The average phase of all 3546 spikes (normalized to 1) is 0.438 (std= 0.118). As mentioned before, this observation agrees with results from the Pulfrich pendulum illusion study [190] which implies a sensitivity to 5 ~ 20 milliseconds.

An aforementioned prerequisite of the multiplexing model is that neurons must be able to modulate their somatic gamma oscillation frequencies. Analyses of the patch-clamp data indicate that gamma frequency is modulated by visual stimuli, as shown in Fig. 5.7a. When the visual stimulus is on, this neuron starts to oscillate more in the gamma frequency range. This observation agrees with GSM's prediction that the selected neuron to code the stimulus will be

assigned a frequency in that range.

When coding the visual stimulus, our modulation hypothesis requires that different processes use separate frequencies. Thus the frequency must change over time to allow a neuron to participate in these processes. Besides, it is unlikely that the gamma frequency would be present for just one cycle but should be stable for several cycles. A standard test for frequency modulation uses the Fourier spectrogram, but this would not have the necessary temporal precision. We locate successive individual zero crossings in the somatic potential and calculate the instantaneous frequencies, as done in Fig. 5.7c. The 20 kHz sampling rate of the data allows for a necessary level of precision.

In Fig. 5.7b, each column shows the frequency histogram from a single trial. The colormap emphasizes the maximums in each trial. The result is that certain gamma frequencies are repeated a significant number of times within the recording periods. These preliminary data are encouraging, as the gamma distributions form discrete peaks (instead of the extremes of being focused in a single frequency across trials or being distributed uniformly). Furthermore, the gamma frequency mode is chosen anew on each trial. In summary, these analyses show that somatic gamma oscillations are modulated is a reasonable assumption.

## 5.6 Discussion, Related Work, and Future Work

Conventional spike recording techniques and modulations by oscillations are typically studied at long temporal scales. This fact obscures the details of

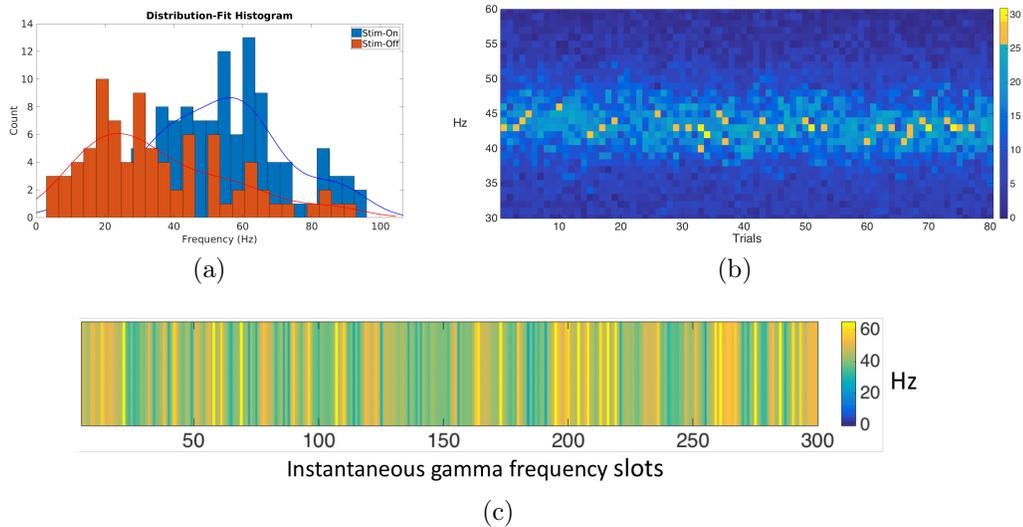


Figure 5.7: (a) Somatic membrane potential oscillations are modulated by visual stimulus. (b) Instantaneous frequency histograms. Each column represents a histogram (bin size = 1Hz) of all the gamma frequencies in a single trial. The color map has been adjusted to accentuate the maxima. The hypothesis is that these values may be indicative of particular individual computations. (c) Taking trial 34 as an example we can ask where the gamma frequencies are in time. This modulation further supports the model as it shows that for a preponderant frequency of 43 Hz there will be many random “slots” that are available to generate a spike for a computation. The results implies a mixing of gamma frequencies as expected by the model.

spike generation that may exploit structure existent at much shorter timescales. This picture is very much echoed by [230], in which one question is particularly germane for this study:

How do microscopic-level neural coding mechanisms interact with macroscopic scale aspects of neural activity?

The focal premise of the paper is that at a fine temporal scale, the cortical neural computation can be factored into independent processes. The model relies crucially on the fine structure of the gamma modulation of a cell's somatic membrane potential, which allows a spike to send a scalar as a delay and be distinguished with a specific gamma frequency. Our model has the following desirable properties:

1. The phase coding mechanism is significantly more efficient than rate coding models. Since each spike can represent an analog quantity, the circuit is not reduced to counting spikes. The speed of a basic clock cycle of a process is that of a gamma-band frequency.
2. Probabilistic selection ensures that neurons in a large population have a chance of being selected and consequently having their receptive fields being updated regularly.
3. One consequence of this model is that a reconciliation of gamma and Poisson action potential observations is straightforward, as shown in the simulation.

4. A neuron can switch its oscillation frequency rapidly, hence multiple processes can time-share this neuron and multiplexing can be achieved.
5. Neurons oscillate in the same frequency can communicate without interference hence multiple processes can coexist.

### 5.6.1 Related Work

First, we elaborate on the assumptions and issues of the three components of our model in the context of related work.

**Phase coding model** The gamma frequency phase code is central to the multiplexing model. With rate coding it is difficult to multiplex and for the downstream neurons to demultiplex the coming spikes. In another study, we show that extracted phase delays from mice viewing oriented sinusoidal grating images are shown to have the same distributions as those from a computer sparse coding model using natural images, suggesting for the first time a direct link between experimentally measured phase delays and model receptive fields [21]. Additionally, previous research has shown that cortical circuits can use spike latency codes [103, 315, 319] regarding a gamma frequency oscillation. The gamma frequency approaches at the network level have demonstrated that long-range communication between distal cells is a reality, in Fries's terms: "communication through coherence" [93].

The proposed model here is similar to the prestigious phase-of-firing model of place cells in the hippocampus with theta frequency oscillation in

the local field potentials (LFP) [224, 285]. Note that this is different from our model since the oscillations in the somatic membrane potentials are used as the reference. This type of model was proposed by [131] but limited to the olfactory and auditory systems. The model was further refined by [19]. Several previous works have suggested that this type of phase coding model is indeed plausible and can coexist with rate code [211, 217, 168].

The Pulfrich pendulum illusion [190] strongly suggests that the cortex is sensitive to delays in the range needed by the model. In a model that depends so completely on exact timing, a very significant problem to address is the mechanisms for writing in and reading out its multiplexed messages. From this perspective, the problem of sensory signal acquisition reduces to one of using phase locking to place the input in register with the phase of the process. A natural site for this to happen is the thalamus, and there is evidence that such mechanisms are used in whisking [342]. Additionally, [168] provides evidence that gamma oscillations at the retina are carried to the cortex through the thalamus.

**Probabilistic parallel neuron selection** The selection algorithm utilizes a simple setting of re-coding image patches from basic (LGN) representations, but a general capability required is that the new coding cells at the next gamma cycle be created from current coding cells calculated. Such an algorithm has been developed [81] and would be usable by the model with some adjustments to make it parallel.

The ability to code a cell's response in one gamma cycle and also to maintain the code on subsequent cycles is important in its own right, but it can also be compatible with the standard *attractor* model of neural computation introduced by [132] and [122], where arbitrary problems can be specified as a fixed point of an appropriate specific network.

**Multiplexing model** At the network level, an issue for larger networks than the one simulated is that the simulation assumes that all neurons that belong to a process are phase-locked. While information traveling between successive neurons can be delayed, there are at least three mitigating circumstances. One is that in the substantially myelinated pyramidal cells, the spike propagation speed is sufficient. The second is that the circuitry between cortical maps may have independent couplings that allow computation to proceed in parallel. The third is that the entire cortex has only ten levels with connectivity arranged in polytrees, potentially allowing the synchronization procedure to settle fast.

In this study, huge simplifications have been made in the course of the exposition of the main results. A major issue that is sidestepped here is that of the control of a process. Multiplexing implies that there must be a method for initiating a process and determining when it is finished. While a complete account of such a mechanism is very much a research topic, rapidly increasing evidence suggests that lower frequencies—theta, alpha, and beta—may play this role [206, 311, 313, 143, 188, 189, 204, 92] as they are shown to modulate gamma frequencies.

### 5.6.2 Implications

Next, we discuss the implications of the GSM model as a whole in the context of previous research.

**Implications for neural signaling** The GSM model provides several new interpretations of cortical processing. Many similar features of the model have been proposed in other contexts [315, 330, 94, 174, 27, 318, 19, 20], but assembling them into a complete system results in a number of innovative features. The proposed model is naturally sympathetic and complementary to the experimental characterizations of the gamma signal [94, 331, 174, 27, 48], but has a different interpretation of the role of the gamma frequency band. In these works, gamma synchronization as a medium makes sense of large-scale communication; gamma synchronization in the small addresses the organization of spike codes in the networks. The GSM model posits that its primary message transmitting role is to allow separate computations to be carried out without crosstalk. In this regard, the model takes a neutral stance as to how the gamma band is sampled, but it may be that this sampling has task-specific components [46].

**Membrane potential issues** The model has finessed several important issues related to the subthreshold membrane potentials. A variety of studies have shown that the membrane potential is implicated in cell responses outside of its classical receptive field, implying that the cells' functional computations

and connectivity are significantly more complex [109, 43, 240]. Their illustrations are significant in themselves but may have additional possibilities in different computations needed by our model and others. One is in the important suggestion for inter-area synchronization by [28]. They propose that the difficult question as to synchronizing feed-forward and feedback spikes may be solved with inter-laminar delays, which could utilize sub-threshold responses that can manipulate the gamma phases [43]. In another issue, our model assumes sub-threshold responses could be used for choosing the cells in a network by binding them to a gamma frequency as well as to prohibiting an action potential for a cell when more than one frequency tries to select it (see Fig 5.3b). Finally, sub-threshold potentials could likely be used in setting general attractor computations such as [132, 252].

**Interpreting receptive fields** The classic interpretation of neural spikes, dating from early work by Barlow [26] is that they belong to a single process and have static interpretations. However, multiplexing changes this interpretation radically and sheds light on an important experiment. In a classical monkey experiment by [213], a V4 receptive field contained two sub-fields A and B. By manipulating the attentional state of the animal, either A or B was attended to, resulting in separate spike rates for each of the foci. However, when the animal is attending away from the receptive field, the response rate is the average of the rates from A and B. From the GSM perspective, when attending to subfield A, the spikes for process A dominate the traffic. But when attending away, A

and B processes could time-share the receptive field hence the spike rate is the average due to multiplexing.

**Interpreting attention** The model also provides an possible interpretation of the effects of attention [202, 203, 42]. This view states that the general effects of attention can be accounted for by a gain factor. The GSM suggests that the gain factor may reflect the use of additional coding neurons by a process. That is, the size of the coding pool is increased directly resulting in an increased probability of any particular neuron being included in a coding process. This leads to improved coding accuracy analogous to the effects that are observed in our sparse coding example.

While attentional effects can be readily measured experimentally, some of the effects appear as small increments on a baseline firing pattern, raising the question of how they can be reliably separated and used [99, 252]. The GSM model has a straightforward answer to this issue in that the attentional effects are captured by separate processes that use separate frequencies. This property is particularly evident in [312] where a large neural transient produced by a transient stimulus gap is immediately ignored afterward. The ability of the neuron immediately to filter out such a large transient signal is readily understood, if the transient and tracing components can be seen as separate processes that share the recorded neuron but use separate gamma frequencies.

### 5.6.3 Future Work: Testing the Model

Eventually, the whole model can be tested precisely once we can perform intracellular recordings of multiple nearby neurons of awake animals. One obvious way is to compare trial averaging and ensemble spectrograms as is done in Fig. 5.6. Power in the gamma range should be much more readily observed in the ensemble recordings. A very exacting test would examine the spectrum for small collections of individual gamma frequencies. This would require estimating the time constants of the coextensive processes, but could be done. The expected result would be that, on a timescale of a few hundred milliseconds, the gamma spectra should be discrete, reflecting the number of independent active processes. A final test may come from advancing techniques for identifying neuron assemblies, which could allow the timing relationships between different assemblies to be analyzed in detail [56].

## 5.7 Conclusion

How does the brain learn and make decisions to achieve behavioral goals in an information-rich environment, with limited cognitive resources?

Theoretical computational models are very important tools for understanding the cortex. In this work we propose the gamma spike multiplexing model that allows neurons to 1) code information efficiently using spike timing, and 2) perform multiple computations in parallel. We use system-level simulations

as well as *in vivo* data to characterize the proposed model. The model can be tested thoroughly once the recording technology becomes available. The proposed model, if proven to be true, would change the way we interpret spike data. At an abstract level, the proposed GSM model provides direct explanations for how modularization and attentional control are realized in the biological brain, which are hypothesized to be important for the brain to learn and act in a complex environment.

How can we improve current artificial intelligence (AI) by studying these mechanisms of the brain, so that AIs can cope with the complexity in the real world?

The proposed GSM model could inspire novel artificial neural network architectures and learning algorithms. The GSM model attempts to answer how modularization and attention are implemented in biological systems. It does not address the question of how such a mechanism is *learned* and formed in the first place. Using a feed-forward artificial neural network as an example, how do certain neurons become specialized to solve a particular task while being trained by a standard back-propagation algorithm [262]? That is, can a group of neurons learn to form modules? If so, how? We conjecture that this could be achieved by utilizing top-down feedback signals, such as reward, to adaptively "recruit" or "dropout" neurons for a particular computation. Another view of this is to perform dropout [291] conditionally based on top-down signals such as task reward instead of randomly – an approach could be named *conditional*

*dropout*. One possible future research direction is to develop a novel neural network model and a learning algorithm that are inspired by the Gamma Spike Multiplexing model. The network model will incorporate feedback connections in addition to the feed-forward connections used by the current neural network models. And one could attempt to develop a learning algorithm to perform conditional dropout in a fashion similar to synaptic gating. The hope is that during the training process, the network will modularize and form relatively independent pathways for different computational tasks.

## Chapter 6

### Conclusion

In this work, the first research question we seek to answer is:

How does the brain learn and make decisions to achieve behavioral goals in an information-rich environment, with limited cognitive resources?

We propose a cognitive model that includes two key components: attention (Chapter 3) and modularization (Chapter 4). We hypothesize that the brain utilizes these two mechanisms in solving complex visuomotor tasks. We have collected two human behavior datasets to test our models: Atari-HEAD<sup>1</sup> and human navigation in VR<sup>2</sup>. To encourage further research in the same direction, we have made these datasets publicly available. We have also proposed a neural coding model, named Gamma Spike Multiplexing (Chapter 5), that explains how attention and modularization mechanisms are implemented in the biological brain. We provide a more complete answer to the above question by discussing the relation between modularization and attention.

---

<sup>1</sup>Available at <https://zenodo.org/record/3451402>

<sup>2</sup>Available at <https://doi.org/10.5281/zenodo.255882>

## 6.1 The Modular Attention Hypothesis

Modularization is closely related to the attention mechanism. An implicit assumption of selective attention is that there exist multiple concurrent cognitive processes so the cortex can multiprocess. Note that our hypothesis does not contradict with the psychological refractory period effect, which suggests a bottleneck encompassing the process of choosing actions and probably memory retrieval generally while performing multiple tasks concurrently [185, 234]. Our hypothesis simply requires that module-specific information need to be held in working memory at the same time.

According to Marr’s paradigm, the highest (most abstract) level of analysis concerns the goal of the computation. Many intelligent behaviors of humans and machines are reward-seeking behaviors, i.e., the goal of the computation is to maximize some reward given a certain task context. In our modular attention model, the overall computational goal is generally to maximize the reward for several ongoing tasks/processes/modules simultaneously. Each module requires specific visual information [289, 259, 307]. In our navigation example, a person must simultaneously follow the path, avoid obstacles, and collect targets. Each of these particular goals requires some visual evaluation of the state of the world to make an appropriate action choice at the moment – and the mechanism that allocates resources (e.g., move one’s eyes) to gather and process needed information for each module is attentional control.

A particular specialization for modules occurs when the action taken

is shared by these modules. A typical case is accounting for the allocation of gaze for sets of modules that have spatially separated targets of interest. Their state estimation is drifting in time with an attendant increase in loss of reward as the policies cannot be as good in the uncertain state estimates. Given that two modules' states cannot be simultaneously updated, which one should get the gaze? A driving study [155] showed that for each module, the expected loss in reward for not looking could be computed, and the module with the highest expected loss should have its state updated with a gaze fixation.

If the above hypothesis turns out to be true, overt attention information can help identify the behavioral modules being activated. A fundamental problem for understanding natural behavior is to be able to predict which modules are currently being considered. Recall in Section 4.3.3 we discussed action selection for modular RL and we hypothesized that the module selection strategy might be more biologically plausible, supported by the previous finding that there might be only one module or a few modules being activated for decision making at a time [155]. Attention information revealed by gaze could be used to infer which module is being executed, hence allow us to better infer the current behavioral goal and predict human actions [112].

Modular RL can potentially reveal the underlying mechanism of attentional control. Estimates of the value of the underlying behaviors will allow the prediction of the gaze patterns subjects make in the environment. It has been suggested that gaze patterns reflect both the subjective value of a target and uncertainty about task-relevant state [289, 155, 307, 105]. For

example, the gaze should be frequently deployed to look at pedestrians in a crowded environment since it is important to avoid collisions and there is high uncertainty about their location. Also, the gaze is deployed very differently depending on the terrain and the need to locate stable footholds, reflecting the increased uncertainty of rocky terrain [201]. Estimates of the subjective value might thus allow inferences about uncertainty as well.

A question remains of where do all these come from in the first place. That is, how do modules/attention/multiple neural processes emerge from the learning experience? The intuition for a modularized strategy comes from two conjectures: learning is incremental and attentional resource is limited. From a developmental perspective, a complicated natural task is often divided into subtasks when learning happens, e.g., curriculum learning [32], hence a real-time decision-making rule is likely to be a combination of pre-learned subroutines. A subtask is attended when needed urgently. Attentional control itself can be treated as a high-level reinforcement learning problem, which can be learned through trial-and-error.

## 6.2 Human-in-the-Loop Reinforcement Learning<sup>3</sup>

Our second research question is:

---

<sup>3</sup>This section of work is based on the following survey paper: Ruohan Zhang, Faraz Torabi, Lin Guan, Dana H Ballard, and Peter Stone. Leveraging human guidance for deep reinforcement learning tasks. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, pages 6339–6346. AAAI Press, 2019. The dissertator is the first author, and takes the leading role in conceiving and designing the survey and writing the paper.

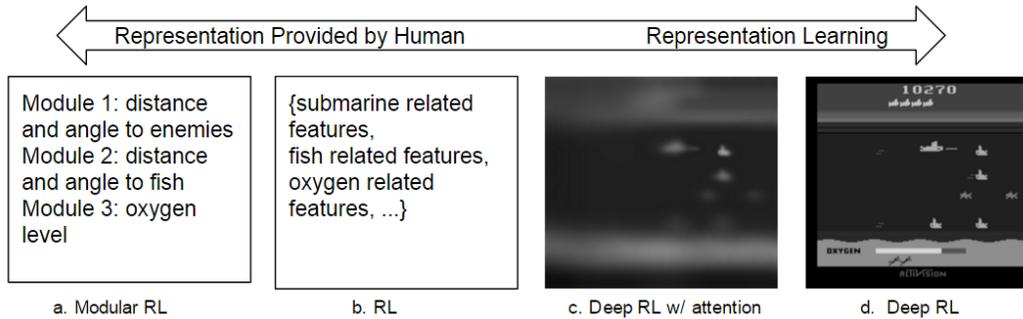


Figure 6.1: The state space representation provided to different reinforcement learning algorithms for Atari Seaquest game.

How can we improve current artificial intelligence (AI) by studying these mechanisms of the brain, so that AIs can cope with the complexity in the real world?

From the perspective of machine learning, representation learning has become a key component in reinforcement learning. Deep RL has successfully bridged representation learning with decision learning. Modularization and attention are both useful inductive biases that alleviate the curse of dimensionality problem for RL agents. They do so by providing more compact state representation (a decomposed state space), or helping the agent learn a better representation (through attention). Fig. 6.1 illustrates the relation between the modular RL, the standard RL, the attention-guided deep RL proposed, and the standard deep RL in terms of how they represent their state space [354]. Modularization can factor a state space into subspaces and can be combined with traditional RL. Visual attention can be readily combined with modern deep RL models.

Our work on modeling human attention and attention-guided learning is well-suited for answering our second research question. Learning attention from human is a novel learning framework which belongs to a broader class of methods that leverages human guidance in training artificial agents (Section 2.3.2). Here we will discuss and compare learning paradigms from five forms of human guidance to provide a broader context for attention learning. These learning paradigms include (1) learning from evaluative feedback (Section 2.3.2.1), (2) learning from human preferences, learning from high-level goals (hierarchical imitation), imitation from observation (Section 2.3.2.2), and learning attention from human [361].

Given the models and notations defined in Chapter 2, diagrams that visualize the interactions between the human trainers, the learning agents, and the task environment for imitation learning together with these five learning frameworks can be found in Fig. 6.1. In (a) standard imitation learning, the human trainer observes state information  $s_t$  and demonstrates action  $a_t^*$  to the agent; the agent stores this data to be used in learning later. In (b) learning from evaluative feedback, the human trainer does not perform the task, instead, he or she watches the agent performing the task, and provides instant feedback  $H_t$  on agent decision  $a_t$  in state  $s_t$ . In (c) learning from human preference. The human trainer watches two behaviors generated by the learning agent simultaneously and decides which behavior is more preferable. In (d) hierarchical imitation, The high-level agent chooses a high-level goal  $g_t$  for state  $s_t$ . The low-level agent then chooses an action  $a_t$  based on  $g_t$

and  $s_t$ . The primary guidance that the trainer provides in this framework is the correct high-level goal  $g_t^*$ . Imitation from observation (e) is similar to standard imitation learning except that the agent does not have access to human demonstrated action – it only observes the state sequence demonstrated by the human. Learning attention from humans (f) requires the trainer to provide attention information  $w_t$  that indicates important task features to the learning agent.

The learning frameworks discussed here are often inspired by real-life biological learning scenarios that correspond to different learning stages and strategies in lifelong learning. Imitation and reinforcement learning correspond to learning completely by imitating others and learning completely through self-generated experience, where the former may be used more often in the early stages of learning and the latter could be more useful in late stages. The other learning strategies discussed are often mixed with these two to allow an agent to utilize signals from all possible sources. For example, it is widely known that children learn largely by imitation and observation [25] at their early stage of learning. Then the children gradually learn to develop joint attention with adults through gaze following [104]. Later children begin to adjust their behaviors based on the evaluative feedback and preference received when interacting with other people. Once they developed the ability to reason abstractly about task structure, hierarchical imitation becomes feasible. At the same time, learning through trial and error from reinforcement is always one of the most common types of learning [286]. The human’s ability to learn

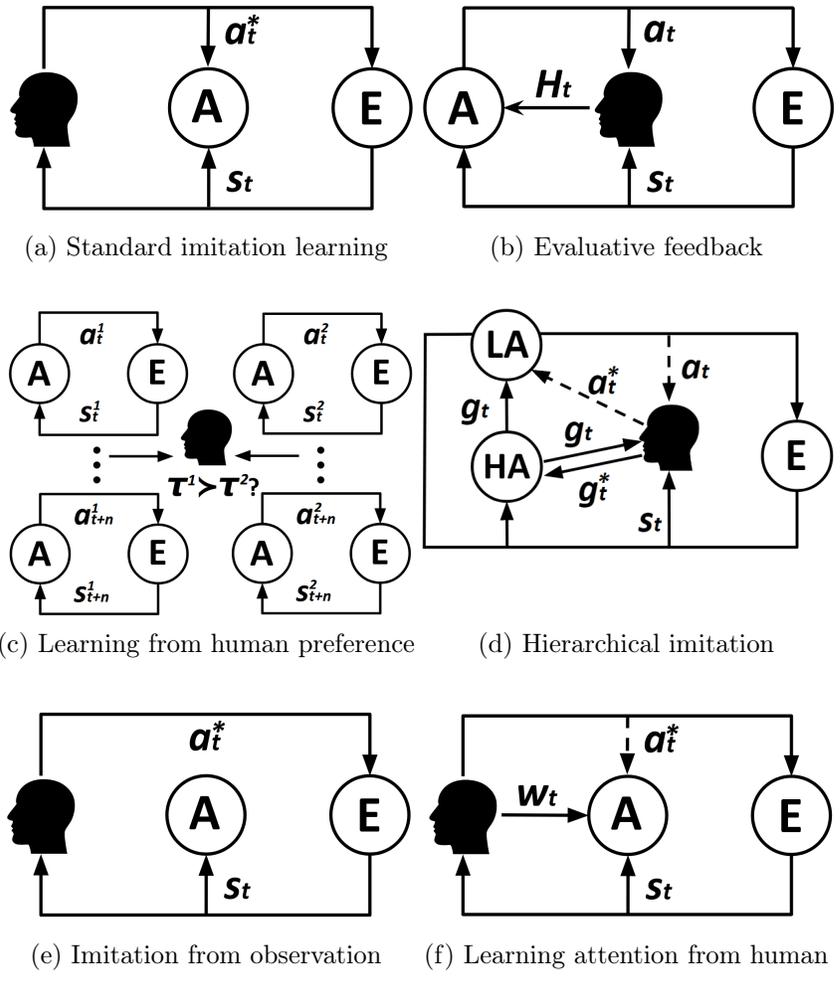


Figure 6.1: Human-agent-environment interaction diagrams of five learning frameworks. These diagrams illustrate how different types of human guidance data are collected, including information required by the human trainer and the guidance provided to the agent. Note that the learning process of the agent is not included in these diagrams. Arrow: information flow direction; Dashed arrow: optional information flow. **A**: learning agent; **E**: environment;  $s_t$ : the state at time  $t$ ;  $a_t$ : agent action. (a)  $a_t^*$ : human demonstrated action. (b)  $H_t$ : human evaluative feedback on agent decision  $a_t$  in state  $s_t$ . (c)  $\tau^1 \succ \tau^2$ : human trainer prefers agent behavior trajectory  $\tau^1$  over  $\tau^2$ . (d) **HA**: a high-level agent that chooses a high-level goal  $g_t$  for state  $s_t$ ; **LA**: a low-level agent that chooses an action  $a_t$  based on  $g_t$  and  $s_t$ ;  $g_t^*$ : high-level goal provided by human. (e) Note that the human demonstrated action  $a_t^*$  is not available to the agent. (f)  $w_t$ : human attention information.

from all types of resources continue to develop through a lifetime.

We have compared these learning strategies within an imitation and reinforcement learning framework. Under this framework, it is possible to develop a unified learning paradigm that accepts multiple types of human guidance. We start to notice efforts towards this goal [4, 326, 102, 333, 218, 35]. In this work, we have seen that human attention learning can be combined with imitation learning (both behavioral cloning and inverse reinforcement learning), learning from evaluative feedback, and imitation from observation. Section 3.5 has shown that incorporating gaze information into imitation from observation and inverse reinforcement learning can lead to a large performance increase in Atari games. Since attention is an intermediate mechanism between perception and action, it becomes very useful when action information is missing in the case of imitation from observation. In learning evaluative feedback and preference, gaze data might reveal more information to the learning agent to explain why

the human gives a particular evaluation. Attention learning is closely related to hierarchical imitation, since gaze is a good indicator of the current high-level behavioral goal which might help an imitator to infer this goal.

### 6.3 Concluding Remarks

We seek to understand and model a modular attention mechanism for humans and animals in various environmental and behavioral settings. Given these reward-seeking visuomotor behaviors, the models attempt to explain the modular attention mechanism at levels II (the representation and algorithm level) and level I (the hardware implementation level) based on David Marr’s paradigm. At the representation and algorithm level, we propose a modular reinforcement learning model for understanding human subjects’ navigation behaviors in an environment with multiple goals. We further develop a modular inverse reinforcement learning algorithm to estimate subjective rewards and discount factors associated with each goal. Also at the representation and algorithm level, we study the active vision problem by jointly modeling human visual attention and actions in video games. Game AIs were shown to benefit from learning human gaze behaviors. We incorporate the learned human attention model into several mainstream imitation and reinforcement learning algorithms. We also compare human attention with RL agent’s attention which allows us to better understand how humans and AIs solve visuomotor tasks differently. At the implementation level, motivated by the modular attention hypothesis, we propose a theoretical neuronal communication model named

gamma spike multiplexing that attempts to explain how the cortex performs multiple computations simultaneously without crosstalk.

## Appendices

# Appendix A

## Attentional Control

### A.1 Atari-HEAD Dataset

Screenshots of 20 Atari games along with eye-tracking data can be found in Fig. A.1. Atari game platform is a rich environment with games of very different dynamics, visual features, and reward functions. Using these games for studying visuomotor control is standard in reinforcement and imitation learning. These games capture many interesting aspects of real-world problems, such as the intercepting task in Breakout and Asterix, driving in Enduro, path planning in Alien, Bank Heist and Ms.Pacman, solving a maze in Hero and Montezuma’s Revenge, and a mixture of tasks in Seaquest and Venture.

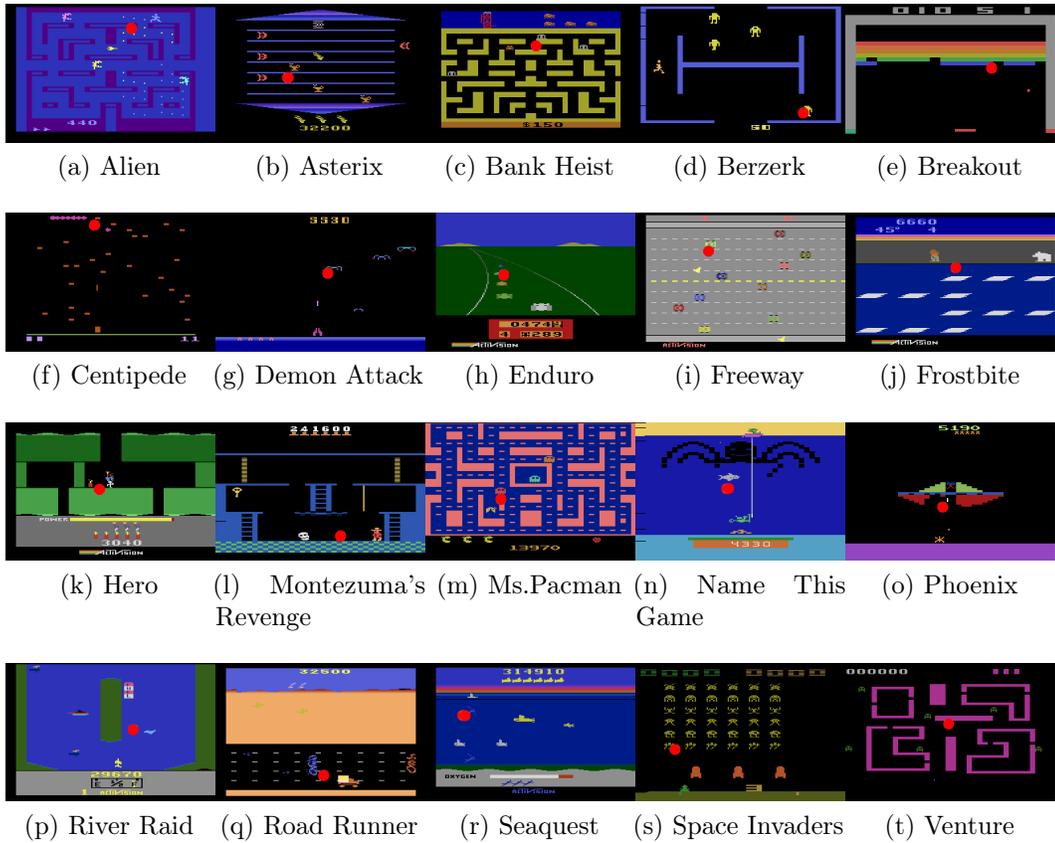


Figure A.1: 20 Atari 2600 games [30] were used to collect human gaze and action data. Red dot indicates human gaze positions.

## A.2 Additional Results for Gaze Modeling

Here we list the details of the gaze prediction network:

- Input image preprocessing: The images are reshaped from  $160 \times 210$  to  $84 \times 84$  with bilinear interpolation and converted into grayscale. Then we scale the pixel values to be in the range of  $[0, 1]$  by dividing them by 255. So the inputs are consistent with the inputs to the reinforcement learning agents.
- Human gaze label preprocessing: Following the convention, we convert discrete gaze positions into continuous distribution by blurring each gaze location using a 2D Gaussian with  $\sigma$  that is equivalent to one visual degree [176, 363].
- Model architecture: The human gaze prediction model is adapted from [363]. The network has three convolution layers followed by three deconvolution layers. Their parameters are as follows:
  - Convolution layer 1: 32 filters, kernel size =  $8 \times 8$ , stride = 4, followed by relu activation, batch normalization, and dropout.
  - Convolution layer 2: 64 filters, kernel size =  $4 \times 4$ , stride = 2, followed by relu activation, batch normalization, and dropout.
  - Convolution layer 3: 64 filters, kernel size =  $3 \times 3$ , stride = 1, followed by relu activation, batch normalization, and dropout.

- Deconvolution layer 1: 64 filters, kernel size =  $3 \times 3$ , stride = 1, followed by relu activation, batch normalization, and dropout.
- Deconvolution layer 2: 64 filters, kernel size =  $4 \times 4$ , stride = 2, followed by relu activation, batch normalization, and dropout.
- Deconvolution layer 3: 1 filter, kernel size =  $8 \times 8$ , stride = 4, followed by a softmax layer.

The network is implemented using Tensorflow 1.8.0 and Keras 2.1.5. The same deep network architecture and hyperparameters are used for all games.

- Optimizer: The optimizer is Adadelta which is a method with adaptive learning rate [346]. We use learning rate = 1.0, decay rate  $\rho = 0.95$ , and  $\epsilon = 1e - 8$ .
- Data: For each game, we use approximately 80% gaze data (16 trials) for training and 20% (4 trials) for testing. For this dataset, two adjacent images or gaze positions are highly correlated. We avoid putting one frame in the training set and its neighboring frame in the testing set by using complete trials as the testing set.
- Hardware: Training was conducted on server clusters with NVIDIA GTX 1080 and 1080Ti GPUs.

Table A.1 shows gaze prediction results for 20 games. For comparison, the performance of the classic bottom-up saliency [144] and optical flow [87]

	Gaze Network				Bottom-up Saliency				Optical Flow			
	NSS	AUC	KL	CC	NSS	AUC	KL	CC	NSS	AUC	KL	CC
alien	6.511	0.973	1.309	0.578	-0.442	0.396	4.714	-0.061	1.093	0.730	8.066	0.115
asterix	4.846	0.966	1.556	0.485	0.104	0.526	4.166	0.001	1.330	0.711	9.959	0.151
bank_heist	6.543	0.974	1.286	0.588	-0.639	0.302	4.511	-0.077	1.669	0.687	11.089	0.161
berzerk	5.280	0.966	1.530	0.503	0.834	0.630	3.903	0.077	1.523	0.646	12.955	0.163
breakout	6.147	0.972	1.266	0.583	-0.047	0.499	4.379	-0.005	2.236	0.665	13.105	0.206
centipede	5.056	0.956	1.750	0.473	0.562	0.673	3.885	0.048	1.276	0.717	11.304	0.131
demon_attack	7.662	0.980	1.084	0.645	-0.247	0.576	4.835	-0.034	1.752	0.764	9.672	0.178
enduro	8.421	0.988	0.830	0.703	-0.248	0.465	4.454	-0.032	0.611	0.728	8.672	0.080
freeway	7.621	0.976	1.133	0.641	-0.158	0.562	4.288	-0.023	1.106	0.700	10.863	0.106
frostbite	5.554	0.961	1.532	0.521	-0.089	0.464	4.346	-0.017	0.625	0.620	12.774	0.072
hero	7.798	0.979	1.061	0.653	0.153	0.554	3.955	0.019	1.893	0.707	11.237	0.195
montezuma	8.267	0.984	0.939	0.683	0.312	0.654	3.816	0.038	1.092	0.684	12.018	0.119
ms_pacman	4.674	0.945	1.858	0.453	-0.380	0.416	4.690	-0.049	1.018	0.668	12.154	0.100
name_this_game	8.164	0.977	1.111	0.653	-0.559	0.367	4.855	-0.069	0.831	0.609	14.039	0.086
phoenix	7.122	0.980	1.153	0.612	-0.256	0.549	4.921	-0.030	1.737	0.742	10.415	0.173
riverraid	6.218	0.966	1.497	0.534	0.063	0.482	4.246	-0.010	1.221	0.727	9.513	0.126
road_runner	6.544	0.973	1.307	0.581	-0.234	0.421	4.227	-0.036	1.626	0.770	8.049	0.170
seaquest	6.350	0.964	1.469	0.552	-0.258	0.345	4.799	-0.042	1.725	0.742	10.147	0.171
space_invaders	6.574	0.982	1.150	0.604	-0.277	0.468	4.758	-0.036	0.847	0.613	14.347	0.087
venture	5.724	0.960	1.605	0.513	0.451	0.608	3.852	0.052	1.110	0.659	12.343	0.114

Table A.1: Quantitative results of predicting human gaze across 20 games. Random prediction baseline: NSS = 0.000, AUC = 0.500, KL = 6.100, CC = 0.000.

models are also computed. A separate convolution-deconvolution network gaze network is trained for each game. The gaze networks are accurate in predicting human gaze (AUC>0.94) for all games.

Fig. A.2 shows the learning curve of the best performing Image+Motion model, where AUC on the testing dataset is plotted against the training data size. The main observation is that training the gaze network for AGIL is sample efficient.

To test whether gaze behavior generalizes across subjects, we show correlation coefficients for the gaze network that is trained using one trial (15-minute) of the data from a single subject and tested on trials from other

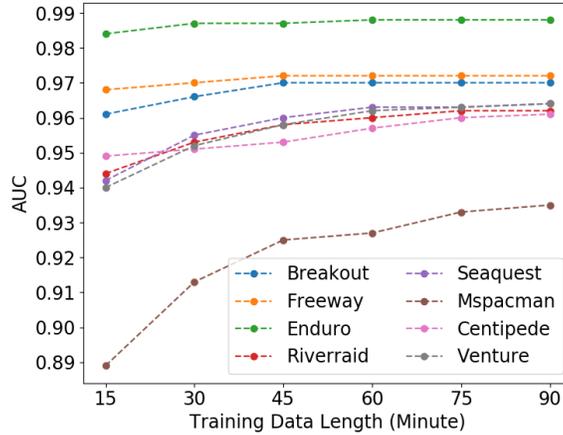


Figure A.2: Gaze prediction learning curve for eight games, i.e., the AUC value obtained when training on the different amounts of data and testing on the same holdout dataset. The model can achieve high AUC values with limited data. Note that we do not show the full scale of the y-axis for better visualization.

subjects (15-minute each). The results are shown in Fig. A.3. In general, the gaze model is more accurate when train and test on the same subjects, indicated by higher correlation coefficient values on the diagonal. The prediction accuracy decreases when training and testing on different subjects by as much as 0.419 (Breakout, trained on Subject3 and tested on Subject1). When tested on a different subject, the average prediction accuracy loss, in terms of the correlation coefficient, is 0.091 compared to trained and tested on the same subject (0.387 vs. 0.478). Meanwhile, the within-subject variance is considerably smaller. When trained on one subject’s data and tested on the same subject, the average mean deviation across subjects and games is 0.011.

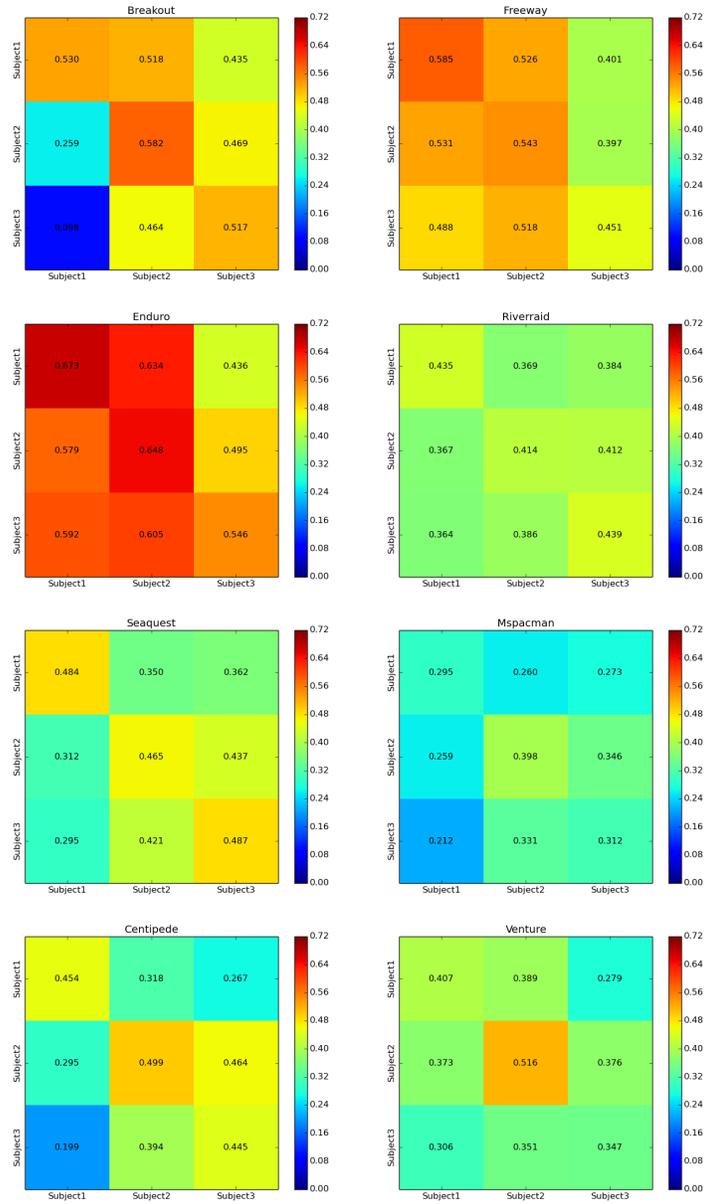


Figure A.3: Correlation coefficient matrices of the gaze network when trained on one subject and tested on another subject. Y axis: Subject ID of the training data. X axis: Subject ID of the testing data.

### A.3 Additional Results for Decision Modeling and Game Playing

In Table A.2 we show behavior matching accuracy of AGIL vs. standard behavioral cloning. In Table A.3 we show game scores of different learning agents using different data. Kurin-IL and Hester-IL are imitation learning results reported in [170] and [125]. The AtariHead-AGIL agent first learns to predict human gaze and uses the learned gaze model to guide the process of learning human decisions. Applying IL and AGIL [359] to our dataset, the mean scores are averaged over 500 episodes per game, with each episode initialized with a randomly generated seed. The game is cut off after 108K frames [124]. The agent chooses an action  $a$  probabilistically using a softmax function with Gibbs (Boltzmann) distribution according to the policy network’s prediction  $P(a)$ :  $\pi(a) = \frac{\exp(\eta P(a))}{\sum_{a' \in \mathcal{A}} \exp(\eta P(a'))}$  where  $\mathcal{A}$  denotes the set of all possible actions,  $\exp(\cdot)$  denotes the exponential function, and the temperature parameter  $\eta$  is set to 1.

Games	Majority baseline	AtariHead-IL	AtariHead-AGIL	Improvement
alien	0.293	0.504	0.690	+0.185
asterix	0.365	0.410	0.532	+0.122
bank_heist	0.278	0.604	0.617	+0.013
berzerk	0.247	0.437	0.482	+0.044
breakout	0.800	0.807	0.816	+0.009
centipede	0.581	0.587	0.628	+0.042
demon_attack	0.316	0.465	0.545	+0.079
enduro	0.406	0.426	0.473	+0.047
freeway	0.781	0.959	0.963	+0.003
frostbite	0.520	0.520	0.639	+0.120
hero	0.483	0.833	0.837	+0.004
montezuma_revenge	0.257	0.866	0.888	+0.023
ms_pacman	0.266	0.555	0.678	+0.123
name_this_game	0.361	0.551	0.746	+0.195
phoenix	0.291	0.574	0.658	+0.084
riverraid	0.339	0.675	0.695	+0.020
road_runner	0.632	0.787	0.809	+0.022
seaquest	0.208	0.414	0.574	+0.160
space_invaders	0.285	0.421	0.505	+0.085
venture	0.196	0.384	0.443	+0.059

Table A.2: Behavior matching accuracy of different models. Majority baseline simply predicts the majority class in that game (the most frequent action). IL: standard imitation learning through behavior cloning. AGIL: policy network that includes saliency map predicted by the gaze network. We also show the improvement of AGIL over standard IL. Random guess prediction accuracy: 0.056.

Games	Kurin-IL	Hester-IL	AtariHead-IL	AtariHead-AGIL	Improvement
alien	-	473.9	1081.5 $\pm$ 741.8	2296.4 $\pm$ 1105.7	+112.33%
asterix	-	279.9	411.5 $\pm$ 192.6	592.4 $\pm$ 290.5	+43.96%
bank_heist	-	95.2	129.3 $\pm$ 75.8	256.1 $\pm$ 116.8	+98.07%
berzerk	-	-	398.0 $\pm$ 189.4	476.6 $\pm$ 197.4	+19.75%
breakout	-	3.5	1.3 $\pm$ 1.4	16.1 $\pm$ 22.5	+1138.46%
centipede	-	-	6169.2 $\pm$ 3856.1	9655.7 $\pm$ 5782.8	+56.51%
demon_attack	-	147.5	2290.4 $\pm$ 1806.7	4465.5 $\pm$ 2603.6	+94.97%
enduro	-	134.8	417.9 $\pm$ 91.4	394.8 $\pm$ 71.2	-5.53%
freeway	-	22.7	30.1 $\pm$ 1.2	30.2 $\pm$ 1.0	+0.33%
frostbite	-	-	2126.6 $\pm$ 1444.3	3233.4 $\pm$ 1857.5	+52.05%
hero	-	5903.3	17134.7 $\pm$ 6454.5	17171.9 $\pm$ 8939.8	+0.22%
montezuma	36 $\pm$ 8.0	576.3	970.2 $\pm$ 896.2	1979.7 $\pm$ 1291.7	+104.05%
ms_pacman	418 $\pm$ 20.0	692.4	1167.5 $\pm$ 686.9	1475.8 $\pm$ 858.5	+26.41%
name_this_game	-	-	5396.6 $\pm$ 1757.0	8557.0 $\pm$ 2015.6	+58.56%
phoenix	-	3745.3	4255.3 $\pm$ 1967.8	6483.3 $\pm$ 3051.5	+52.36%
riverraid	-	2148.5	2639.6 $\pm$ 669.3	4106.4 $\pm$ 1457.1	+55.57%
road_runner	-	8794.9	28311.2 $\pm$ 7261.8	42539.4 $\pm$ 11177.2	+50.26%
seaquest	144 $\pm$ 12.4	195.6	205.6 $\pm$ 103.7	841.0 $\pm$ 842.1	+309.05%
space_invaders	-	-	247.1 $\pm$ 149.2	248.2 $\pm$ 147.1	+0.45%
venture	-	-	286.0 $\pm$ 146.8	400.0 $\pm$ 175.4	+39.86%

Table A.3: Game scores (mean  $\pm$  standard deviation) of game agents trained using different sources of data. The scale and quality of our data leads to better performance, when comparing to AtariHEAD-IL to Kurin-IL and Hester-IL. Incorporating attention leads to an average improvement of 115.26% over a standard IL algorithm using our dataset.

## A.4 Additional Results for Coverage-Based Gaze Loss

Here we show game scores for behavioral cloning (Table A.4 and A.5), behavioral cloning from observation (Table A.6 and A.7), and T-REX (Table A.8 and A.9). We also show the results of reducing causal confusion with CGL (Table A.10).

	BC	BC-2ch	AGIL	BC+CGL	Improv-BC-2ch	Improv-AGIL	Improv-CGL
alien	1575±176.8	1296.7±140.8	1866.7±171.3	<b>2044.7±242.1</b>	-17.7%	18.5%	29.8%
asterix	285±28.2	283.3±31.1	275±35.9	<b>426.7±27.8</b>	-0.6%	-3.5%	49.7%
bank_heist	86.3±9.0	129.3±17.7	<b>169±13.1</b>	143±14.8	49.8%	95.8%	65.7%
berzerk	330.7±22.0	350±22.5	251.7±19.1	<b>366.7±19</b>	5.8%	-23.9%	10.9%
breakout	2.2±0.3	2.7±0.3	<b>4.9±0.4</b>	3.7±0.4	22.7%	122.7%	68.2%
centipede	4378.8±442.6	5762.1±687.6	4600.2±333.8	<b>6075.9±845.1</b>	31.6%	5.1%	38.8%
demon_attack	112.2±13.9	177±18.7	148±16.9	<b>205.2±41.9</b>	57.8%	31.9%	82.9%
enduro	<b>11.7±2.0</b>	7.8±1.4	0±0	4.2±1.4	-33.3%	-100.0%	-64.1%
freeway	29.4±0.2	28.5±0.3	28.1±0.3	<b>30±0.3</b>	-3.1%	-4.4%	2.0%
frostbite	1628.3±246.4	1406.3±266.5	<b>3185±352.9</b>	2973±279	-13.6%	95.6%	82.6%
hero	13255.3±845.1	18877.2±509.0	15582.7±789.7	<b>19023.2±679.7</b>	42.4%	17.6%	43.5%
montezuma_revenge	100±31.6	0±0.0	0±0	<b>1200±159.2</b>	-100.0%	-100.0%	1100.0%
ms_pacman	843.3±62.8	783.3±55.8	1072.3±74.8	<b>1348.3±206.9</b>	-7.1%	27.2%	59.9%
name_this_game	1917.3±130.2	2153.3±169.1	<b>2832±194.2</b>	2646.3±156.3	12.3%	47.7%	38.0%
phoenix	1060±172.4	1105±159.7	1171.7±147.6	<b>2193.7±200.5</b>	4.2%	10.5%	107.0%
riverraid	2771.7±141.8	2701.3±128.0	<b>3900.3±223.6</b>	2965.3±184.8	-2.5%	40.7%	7.0%
road_runner	7840±553.3	3820±372.3	6920±518.8	<b>12723.3±376.7</b>	-51.3%	-11.7%	62.3%
seaquest	194±11.4	162±9.8	198±12.7	<b>216±11.2</b>	-16.5%	2.1%	11.3%
space_invaders	275±26.2	275±29.3	254.7±21.1	<b>314±26</b>	0.0%	-7.4%	14.2%
venture	196.7±26.5	273.3±27.1	73.3±24.9	<b>376.7±16.1</b>	38.9%	-62.7%	91.5%
average	-	-	-	-	1.0%	10.1%	<b>95.1%</b>

Table A.4: Game scores obtained when using 15-minute human demonstration data to train the behavioral cloning agents. Results are presented as mean±standard error of the mean (N=30). The agents we compare are behavioral cloning agent (BC), two channeled behavioral cloning agent (BC-2ch), Attention-guided imitation learning agent (AGIL), and proposed coverage-based loss agent (CGL). The improvement columns show the relative improvement over the BC baseline.

	BC	BC-2ch	AGIL	BC+CGL	Improv-BC-2ch	Improv-AGIL	Improv-CGL
alien	694±97.5	1303.7±147.6	<b>2104.7±180.2</b>	2027.3±140.1	87.9%	203.3%	192.1%
asterix	516.7±54.9	465±32	455±50.1	<b>773.3±63.4</b>	-10.0%	-11.9%	49.7%
bank_heist	102.3±8	<b>174.3±13.9</b>	117.3±13.3	156.3±12.5	70.4%	14.7%	52.8%
berzerk	256.7±17.4	379.3±29.3	188.3±24.5	<b>435±42.3</b>	47.8%	-26.6%	69.5%
breakout	1.7±0.3	<b>3.6±0.4</b>	<b>3.6±0.4</b>	2.9±0.3	111.8%	111.8%	70.6%
centipede	7704.5±967	7856.8±903.1	9073.7±914.8	<b>9330±922</b>	2.0%	17.8%	21.1%
demon_attack	848.2±140.3	333.8±38.6	<b>2156.2±295.2</b>	1375±216.7	-60.6%	154.2%	62.1%
enduro	386±12.1	385.4±12	278.7±17.5	<b>445.1±17</b>	-0.2%	-27.8%	15.3%
freeway	27.6±0.3	<b>31.4±0.1</b>	28.9±0.3	30.2±0.2	13.8%	4.7%	9.4%
frostbite	2016.7±180.2	2331.7±240.4	1980±176.4	<b>3253±254.5</b>	15.6%	-1.8%	61.3%
hero	9519.7±874.9	11152±1079.9	7685.7±1100.6	<b>16936.5±1342.6</b>	17.1%	-19.3%	77.9%
montezuma_revenge	490±109	1480±168.8	553.3±70.3	<b>1720±156.3</b>	202.0%	12.9%	251.0%
ms_pacman	1200±123.2	1511±144.1	1272.3±128.6	<b>1590±197.1</b>	25.9%	6.0%	32.5%
name_this_game	2887±177.8	5732±288.2	<b>5817±341.7</b>	5405.3±267	98.5%	101.5%	87.2%
phoenix	4029±279.8	4597.3±324.1	<b>5140±405.2</b>	4472.7±440.2	14.1%	27.6%	11.0%
riverraid	2806±164.5	2266.3±65.7	2555.7±56.2	<b>3452.7±242.1</b>	-19.2%	-8.9%	23.0%
road_runner	29433.3±1230.3	31776.7±1426.2	29410±1516.2	<b>33510±1026.3</b>	8.0%	-0.1%	13.9%
seaquest	175.5±30.7	182.1±12.4	443±124.6	<b>610.7±106.8</b>	3.8%	152.4%	248.0%
space_invaders	243.8±26.7	196±26.4	206±23	<b>363.5±35.7</b>	-19.6%	-15.5%	49.1%
venture	73.3±27.9	43.3±20.9	<b>330±21.7</b>	313.3±29	-40.9%	350.2%	327.4%
average	-	-	-	-	28.4%	52.3%	<b>86.2%</b>

Table A.5: Game scores obtained when using all 300-minute human demonstration data to train the behavioral cloning agents. Results are presented as mean±standard error of the mean (N=30). The agents we compare are behavioral cloning agent (BC), two channeled behavioral cloning agent (BC-2ch), Attention-guided imitation learning agent (AGIL), and proposed coverage-based loss agent (CGL). The improvement columns show the relative improvement over the BC baseline.

	BCO	BCO+GMD	BCO+Motion	BCO+CGL	Improv-GMD	Improv-Motion	Improvement-CGL
alien	0.0 ± 0.0	<b>140.00 ± 0.00</b>	<b>140.00 ± 0.00</b>	<b>140.00 ± 0.00</b>	-	-	-
asterix	288.33 ± 30.48	645.00 ± 33.28	<b>700.00 ± 0.00</b>	253.33 ± 9.11	123.70%	142.80%	-12.10%
bank_heist	0.0 ± 0.0	0.00 ± 0.00	<b>8.67 ± 0.62</b>	0.00 ± 0.00	0%	-	0%
berzerk	158.33 ± 16.45	263.67 ± 26.36	528.33 ± 17.11	<b>687.67 ± 27.98</b>	66.50%	233.70%	334.30%
breakout	0.0 ± 0.0	<b>2.30 ± 0.08</b>	<b>2.30 ± 0.08</b>	0.60 ± 0.17	-	-	-
centipede	646.83 ± 79.70	2707.83 ± 309.30	3234.13 ± 167.51	<b>5047.93 ± 410.74</b>	318.60%	400%	680.41%
demon_attack	157.33 ± 20.87	<b>844.00 ± 88.40</b>	806.00 ± 85.25	791.83 ± 77.56	436.50%	412.30%	403.30%
enduro	0.0 ± 0.0	0.13 ± 0.13	1.57 ± 0.72	<b>7.77 ± 0.76</b>	-	-	-
freeway	0.0 ± 0.0	<b>21.30 ± 0.21</b>	<b>21.30 ± 0.21</b>	<b>21.30 ± 0.21</b>	-	-	-
frostbite	116.33 ± 6.86	79.33 ± 2.94	80.67 ± 6.43	<b>160.00 ± 0.00</b>	-31.80%	-30.70%	37.50%
hero	0.0 ± 0.0	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0%	0%	0%
montezuma_revenge	0.0 ± 0.0	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0%	0%	0%
ms_pacman	60 ± 0	<b>379.00 ± 11.27</b>	210.00 ± 0.00	210.00 ± 0.00	531.70%	250%	250%
name_this_game	694.67 ± 86.88	2183.00 ± 52.93	<b>2770.00 ± 0.00</b>	<b>2770.00 ± 0.00</b>	214.20%	298.80%	298.80%
phoenix	282 ± 28.95	352.67 ± 29.26	<b>407.33 ± 66.01</b>	256.67 ± 22.43	25.10%	44.40%	-9%
riverraid	360 ± 0	1029.33 ± 13.48	236.00 ± 10.93	<b>1250.00 ± 0.00</b>	185.90%	-34.40%	247.20%
road_runner	0.0 ± 0.0	473.33 ± 77.16	500.00 ± 91.29	<b>956.67 ± 9.05</b>	-	-	-
seaquest	0.0 ± 0.0	102.00 ± 4.46	<b>140.00 ± 0.00</b>	<b>140.00 ± 0.00</b>	-	-	-
space_invaders	220.83 ± 27.76	195.17 ± 17.64	<b>285.00 ± 0.00</b>	270.00 ± 0.00	-11.60%	29.10%	22.30%
venture	0.0 ± 0.0	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0%	0%	0%
average	-	-	-	-	132.8%	134.3%	160.9%

Table A.6: Game scores obtained when using 15-minute human demonstration data to train the agents. Results are presented as mean±standard error of the mean (N=30). The agents we compare are behavioral cloning from observation agent (BCO), gaze-modulated dropout (GMD), BCO with motion information, and BCO+CGL. The improvement columns show the relative improvement over the BCO baseline. “-” indicates that the baseline score is zero hence the relative improvement is not calculated and is not counted in the average.

	BCO	BCO+GMD	BCO+Motion	BCO+CGL	Improv-GMD	Improv-Motion	Improvement-CGL
alien	140.00 ± 0.00	140.00 ± 0.00	140.00 ± 0.00	140.00 ± 0.00	0%	0%	0%
asterix	181 ± 13.02	276.67 ± 24.24	650.00 ± 0.00	<b>690.00 ± 27.33</b>	52.90%	259.10%	281.20%
bank_heist	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0%	0%	0%
berzerk	145.00 ± 15.87	229.00 ± 22.40	539.00 ± 22.99	<b>572.33 ± 17.71</b>	57.90%	271.70%	294.70%
breakout	0.3 ± 0.08	2.00 ± 0.00	0.60 ± 0.17	0.17 ± 0.07	566.60%	100%	-43.30%
centipede	184 ± 0.91	3309.03 ± 186.34	3668.27 ± 192.32	<b>8391.03 ± 557.52</b>	1698.40%	1893.60%	4460.30%
demon_attack	127.67 ± 19.33	180.00 ± 18.60	806.00 ± 85.25	806.00 ± 85.25	41.00%	531.30%	531.30%
enduro	2.93 ± 0.81	0.37 ± 0.17	0.07 ± 0.05	0.00 ± 0.00	-87.20%	-97.60%	-100%
freeway	0.00 ± 0.00	21.30 ± 0.21	21.30 ± 0.21	21.30 ± 0.21	-	-	-
frostbite	102.33 ± 6.14	329.67 ± 53.65	<b>160.00 ± 0.00</b>	124.00 ± 12.68	222.20%	56.40%	21.20%
hero	0.00 ± 0.00	150.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-	0%	0%
montezuma_revenge	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0%	0%	0%
ms_pacman	60 ± 0	210.00 ± 0.00	210.00 ± 0.00	210 ± 0	250%	250%	250%
name_this_game	1158.00 ± 50.88	1808.00 ± 75.44	2770.00 ± 0.00	2770.00 ± 0.00	56.10%	139.20%	139.20%
phoenix	147.33 ± 6.22	444.00 ± 52.35	<b>474.00 ± 75.41</b>	356.43 ± 70.41	201.40%	221.70%	141.90%
riverraid	360.00 ± 0.00	1646.33 ± 50.27	440.00 ± 0.00	<b>1222.00 ± 3.98</b>	357.30%	22.20%	239.40%
road_runner	0.00 ± 0.00	493.33 ± 97.52	<b>956.67 ± 9.05</b>	0.00 ± 0.00	-	-	0%
seaquest	0.00 ± 0.00	120.00 ± 0.00	180.00 ± 0.00	180 ± 0	-	-	-
space_invaders	<b>398.00 ± 16.65</b>	278.00 ± 11.71	285.00 ± 0.00	273.17 ± 11.93	-30.20%	-28.40%	-31.40%
venture	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0%	0%	0%
average	-	-	-	-	211.7%	212.9%	343.6%

Table A.7: Game scores obtained when using 300-minute human demonstration data to train the agents. Results are presented as mean±standard error of the mean (N=30). The agents we compare are behavioral cloning from observation agent (BCO), gaze-modulated dropout (GMD), BCO with motion information, and BCO+CGL. The improvement columns show the relative improvement over the BCO baseline. "-" indicates that the baseline score is zero hence the relative improvement is not calculated and is not counted in the average.

	T-REX	T-REX +CGL	Improv-CGL
alien	727.00 ± 52.24	<b>800.33 ± 66.44</b>	10.1%
asterix	5023.33 ± 431.17	<b>66445.00 ± 7444.18</b>	1222.7%
bank_heist	0.00 ± 0.00	<b>19.33 ± 3.86</b>	-
berzerk	273.33 ± 10.20	<b>584.00 ± 22.18</b>	113.7%
breakout	46.33 ± 1.75	<b>386.77 ± 25.42</b>	734.8%
centipede	8369.30 ± 971.79	<b>20762.47 ± 2120.94</b>	148.1%
demon_attack	<b>64.00 ± 5.18</b>	0.00 ± 0.00	-100.0%
enduro	0.00 ± 0.00	0.00 ± 0.00	0.0%
freeway	0.00 ± 0.00	<b>0.07 ± 0.05</b>	-
frostbite	1.00 ± 0.55	<b>36.67 ± 1.28</b>	3567.0%
hero	0.00 ± 0.00	0.00 ± 0.00	0.0%
montezuma_revenge	0.00 ± 0.00	0.00 ± 0.00	0.0%
ms_pacman	<b>967.00 ± 84.22</b>	577.33 ± 61.54	-40.3%
name_this_game	2262.00 ± 104.89	<b>4081.00 ± 175.34</b>	80.4%
phoenix	303.67 ± 28.24	<b>502.67 ± 39.24</b>	65.5%
riverraid	1748.00 ± 31.49	<b>5201.67 ± 203.35</b>	197.6%
road_runner	0.00 ± 0.00	<b>2660.00 ± 359.77</b>	-
seaquest	0.00 ± 0.00	<b>759.33 ± 12.56</b>	-
space_invaders	607.00 ± 44.32	<b>923.50 ± 59.82</b>	52.1%
venture	0.00 ± 0.00	0.00 ± 0.00	0.0%
average	-	-	390.4%

Table A.8: Game scores obtained when using 30-minute human demonstration data to train the T-REX agents. Results are presented as mean±standard error of the mean (N=30). The improvement columns show the relative improvement over the T-REX baseline. “-” indicates that the baseline score is zero hence the relative improvement is not calculated and is not counted in the average.

	T-REX	T-REX +CGL	Improv-CGL
alien	359.67 ± 8.48	<b>1007.33 ± 48.94</b>	180.10%
asterix	15231.67 ± 2401.26	<b>17073.33 ± 2253.10</b>	12.10%
bank_heist	2.33 ± 0.77	<b>7.00 ± 1.17</b>	200.40%
berzerk	411.67 ± 40.87	<b>596.67 ± 32.52</b>	44.90%
breakout	53.33 ± 1.30	<b>438.40 ± 17.59</b>	722.10%
centipede	<b>16363.07 ± 1993.35</b>	13532.70 ± 1550.44	-17.30%
demon_attack	463.17 ± 138.3	<b>17589.00 ± 1727.02</b>	3697.50%
enduro	<b>0.90 ± 0.57</b>	0.67 ± 0.16	-25.60%
freeway	0.00 ± 0.00	<b>0.07 ± 0.05</b>	-
frostbite	22.67 ± 1.88	<b>208.00 ± 6.28</b>	817.50%
hero	0.00 ± 0.00	<b>2.50 ± 2.46</b>	-
montezuma_revenge	0.00 ± 0.00	0.00 ± 0.00	0%
ms_pacman	314.00 ± 22.41	<b>527.33 ± 38.74</b>	67.90%
name_this_game	3331.67 ± 185.85	<b>4010.67 ± 147.50</b>	20.40%
phoenix	<b>4322.33 ± 363.98</b>	2123.67 ± 164.78	-50.90%
riverraid	5812.67 ± 233.48	<b>7370.00 ± 262.18</b>	26.80%
road_runner	0.00 ± 0.00	<b>1286.67 ± 111.73</b>	-
seaquest	0.00 ± 0.00	<b>729.33 ± 16.32</b>	-
space_invaders	410.50 ± 46.54	<b>1563.67 ± 144.76</b>	280.90%
venture	0.00 ± 0.00	0.00 ± 0.00	0%
average	-	-	373.6%

Table A.9: Game scores obtained when using 300-minute human demonstration data to train the T-REX agents. Results are presented as mean±standard error of the mean (N=30). The improvement columns show the relative improvement over the T-REX baseline. “-” indicates that the baseline score is zero hence the relative improvement is not calculated and is not counted in the average.

	BC	CGL	Improv-CGL	BC-confounded	CGL-confounded	Improv-CGL-confounded	Change-BC	Change-CGL
alien	1575±176.8	2044.7±242.1	29.8%	73±9.3	439.3±63.4	501.8%	-95.4%	-78.5%
asterix	285±28.2	426.7±27.8	49.7%	243.3±23.1	363.3±30	49.3%	-14.6%	-14.9%
bank_heist	86.3±9.0	143±14.8	65.7%	22.3±3.2	15.3±2.8	-31.4%	-74.2%	-89.3%
berzerk	330.7±22.0	366.7±19	10.9%	101.7±11.2	322±21.5	216.6%	-69.2%	-12.2%
breakout	2.2±0.3	3.7±0.4	68.2%	0±0	0.4±0.1	-	-100.0%	-89.2%
centipede	4378.8±442.6	6075.9±845.1	38.8%	5320.8±705.5	5808.5±640.5	9.2%	21.5%	-4.4%
demon_attack	112.2±13.9	205.2±41.9	82.9%	120.5±10	200.7±32.1	66.6%	7.4%	-2.2%
enduro	11.7±2.0	4.2±1.4	-64.1%	3.3±0.7	8.3±1.5	151.5%	-71.8%	97.6%
freeway	29.4±0.2	30±0.3	2.0%	23.9±0.2	26.5±0.2	10.9%	-18.7%	-11.7%
frostbite	1628.3±246.4	2973±279	82.6%	189.3±3.7	710±99.3	275.1%	-88.4%	-76.1%
hero	13255.3±845.1	19023.2±679.7	43.5%	109.7±97.8	10038.3±647.6	9050.7%	-99.2%	-47.2%
montezuma_revenge	100±31.6	1200±159.2	1100.0%	0±0	0±0	0.0%	-100.0%	-100.0%
ms_pacman	843.3±62.8	1348.3±206.9	59.9%	281±65.4	397±31	41.3%	-66.7%	-70.6%
name_this_game	1917.3±130.2	2646.3±156.3	38.0%	2204.7±196.1	3179.3±173.4	44.2%	15.0%	20.1%
phoenix	1060±172.4	2193.7±200.5	107.0%	550.3±76.8	1657±263.9	201.1%	-48.1%	-24.5%
riverraid	2771.7±141.8	2965.3±184.8	7.0%	2457±113.8	2105±67.1	-14.3%	-11.4%	-29.0%
road_runner	7840±553.3	12723.3±376.7	62.3%	6290±364.1	10023.3±396	59.4%	-19.8%	-21.2%
seaquest	194±11.4	216±11.2	11.3%	182.7±10.3	182±10.1	-0.4%	-5.8%	-15.7%
space_invaders	275±26.2	314±26	14.2%	219.3±21.9	265.5±26.9	21.1%	-20.3%	-15.4%
venture	196.7±26.5	376.7±16.1	91.5%	6.7±6.6	20±11	198.5%	-96.6%	-94.7%
average	-	-	95.1%	-	-	571.1%	-47.8%	-34.0%

Table A.10: Causal confusion study results. Game scores are obtained when using 15-minute human demonstration data to train the agents. Results are presented as mean±standard error of the mean (N=30). The change columns show the relative change over the non-confounded baselines. On average CGL agents suffer less when trained with confounded data, and still perform better than behavioral cloning (BC) agents.

## A.5 Additional Results for Human versus Machine Attention

In the following sections, we will show statistics and example images of each game. Atari games have very different reward mechanisms, visual features, and dynamics. Hence, it is often difficult to find an RL algorithm that works best for all games. We hope that, by showing results for individual games, researchers will gain insights into why a particular algorithm (like PPO here) performs well or poorly for a particular game.

### A.5.1 The Effects of Learning on Attention

In Fig. A.4 to A.9 we show how the attention of the RL agent (PPO) evolves over time compared to human attention. Part (a) of each figure shows the similarity metrics: Pearson’s Correlation Coefficient (CC) and negative Kullback-Leibler Divergence (KL) values over training time steps. The values are averaged over 100 images in the standard image set (as described in section 3.3). (a) also shows the game scores (averaged over 50 episodes) over training time steps. Part (b) of each figure shows an example game image. It also includes the average saliency maps of the RL agents during training and a human saliency map predicted by the human model for the selected game image. Note that KL values are negated for better visualization.

### A.5.2 The Effects of Discount Factors on Attention

Fig. A.10 to A.15 shows how the attention of the RL agent (PPO) changes when we vary the discount factor, compared to human attention.  $\gamma = 0.99$  is the default value for most RL algorithms [127, 269]. Each Figure (a) shows the similarity metrics: CC and negative KL values over different discount factors. The values are averaged over 100 images in the standard image set. (a) also shows the game scores (averaged over 50 episodes) over discount factors. Each Figure (b) shows an example game image, RL agents' saliency maps with different discount factors  $\gamma$ , and human saliency map predicted by the human model. Note that KL values are negated for better visualization.

### A.5.3 Failure States Analysis

Fig. A.16 and A.17 show RL agents' saliency maps compared to human's in failure states. These states are game frames right before the RL agent loses a "life" which incurs a large penalty in Atari games. This analysis helps answer the question: Did RL agents make mistakes because they fail to attend to the right objects, or did they attend to the right objects but make wrong decisions? Figure 13 shows the games that belong to the former case, and Figure 14 shows the games that belong to the latter case. Freeway is excluded here since the PPO agent learned a nearly optimal policy.

#### A.5.4 Generalizing to Unseen Data

Fig. A.18 and A.19 show RL agents' saliency maps compared to human's in unseen states. The unseen states are end-game states obtained from human experts' data which RL agents have not encountered during learning. The goal is to see whether RL agents' attention can reasonably generalize to these unseen states. Again, Freeway is excluded here since the PPO agent learned a nearly optimal policy.

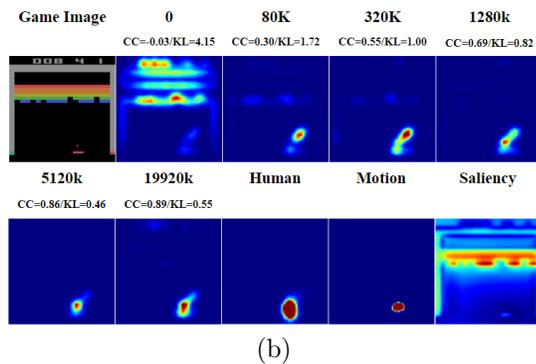
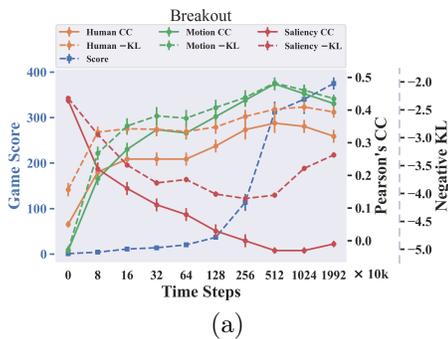


Figure A.4: Breakout: (a) Human and RL saliency maps become more similar over training time steps. Pearson’s correlation coefficients between game score and human are CC:  $r(8) = 0.664, p < 0.05$ , KL:  $r(8) = 0.622, p = 0.054$ ; between game score and motion are CC:  $r(8) = 0.641, p < 0.05$ , KL:  $r(8) = 0.550, p = 0.100$ ; between game score and saliency are CC:  $r(8) = -0.661, p < 0.05$ , KL:  $r(8) = -0.215, p = 0.551$ . (b) The RL agents gradually learn to focus their attention on both the paddle and the ball as humans do.

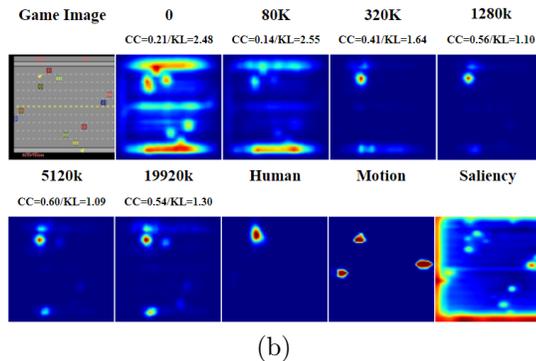
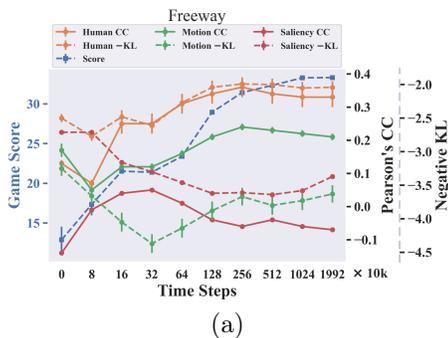
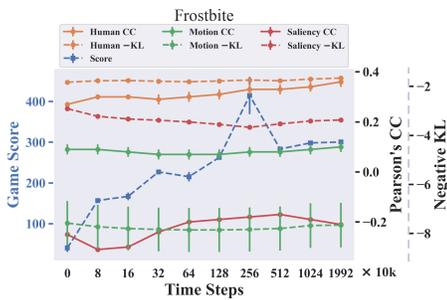
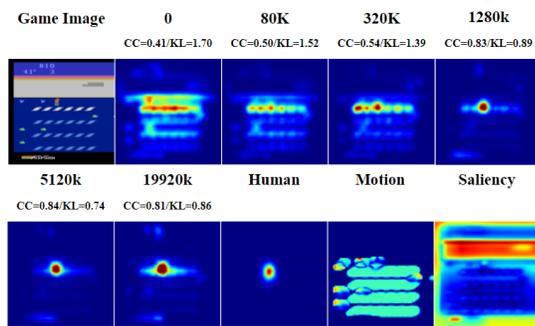


Figure A.5: Freeway: (a) Human and RL saliency maps become more similar over training time steps. Pearson’s correlation coefficients between game score and human are CC:  $r(8) = 0.878, p < 0.001$ , KL:  $r(8) = 0.888, p < 0.001$ ; between game score and motion are CC:  $r(8) = 0.765, p < 0.01$ , KL:  $r(8) = 0.080, p = 0.826$ ; between game score and saliency are CC:  $r(8) = -0.078, p = 0.830$ , KL:  $r(8) = -0.878, p < 0.001$ . (b) The RL agents gradually learn to focus their attention on the yellow chicken being controlled to cross the highway. The similarity values decrease a little at the end of the training because the RL agents also learn to attend to the starting point at the bottom of the image.



(a)



(b)

Figure A.6: Frostbite: (a) Human and RL saliency maps become more similar over training time steps. Pearson’s correlation coefficients between game score and human are CC:  $r(8) = 0.791, p < 0.01$ , KL:  $r(8) = 0.620, p = 0.056$ ; between game score and motion are CC:  $r(8) = -0.087, p = 0.811$ , KL:  $r(8) = -0.443, p = 0.200$ ; between game score and saliency are CC:  $r(8) = 0.688, p < 0.05$ , KL:  $r(8) = -0.900, p < 0.001$ . (b) The RL agents gradually learn to attend to the little person being controlled in the middle like humans do.

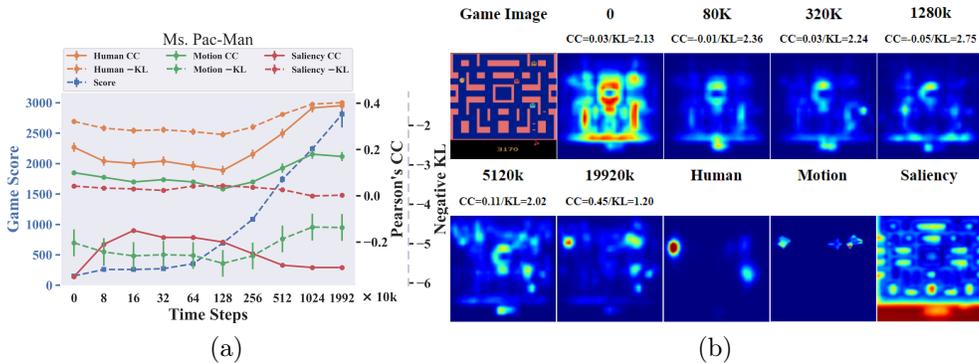


Figure A.7: Ms.Pac-Man: (a) Human and RL saliency maps become less similar at first, and then become more similar during training. Pearson's correlation coefficients between game score and human are CC:  $r(8) = 0.910, p < 0.001$ , KL:  $r(8) = 0.893, p < 0.001$ ; between game score and motion are CC:  $r(8) = 0.819, p < 0.01$ , KL:  $r(8) = 0.809, p < 0.01$ ; between game score and saliency are CC:  $r(8) = -0.586, p = 0.075$ , KL:  $r(8) = -0.806, p < 0.01$ . (b) The RL agents eventually learn to attend to the Pac-Man on the left and an enemy ghost on the right like humans do.

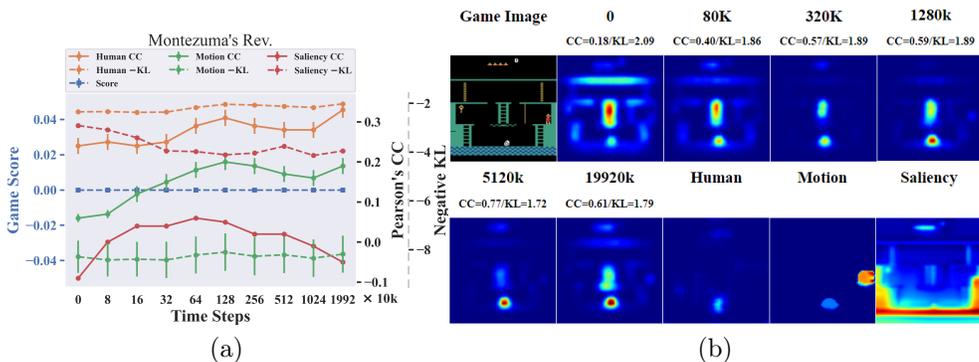


Figure A.8: Montezuma's Revenge: (a) Human and RL saliency maps becomes more similar over training time steps. Note that this is a difficult game for RL agents and they never learn to score. Pearson's correlation coefficients are undefined in this case. (b) The RL agents learn to attend to the enemy at the bottom like humans do, but they are uncertain about the importance of the ladder in the middle.

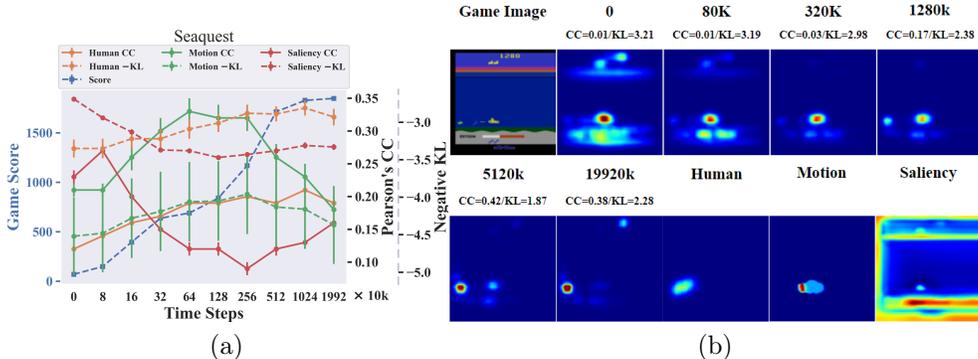


Figure A.9: Seaquest: (a) Human and RL saliency maps becomes more similar according to the KL metric. Pearson’s correlation coefficients between game score and human are CC:  $r(8) = 0.824, p < 0.01$ ; KL:  $r(8) = 0.930, p < 0.001$ ; between game score and motion are CC:  $r(8) = -0.116, p = 0.750$ , KL:  $r(8) = 0.419, p = 0.228$ ; between game score and saliency are CC:  $r(8) = -0.653, p < 0.05$ , KL:  $r(8) = -0.641, p < 0.05$ . (b) The RL agents learn to attend to an incoming enemy on the left.

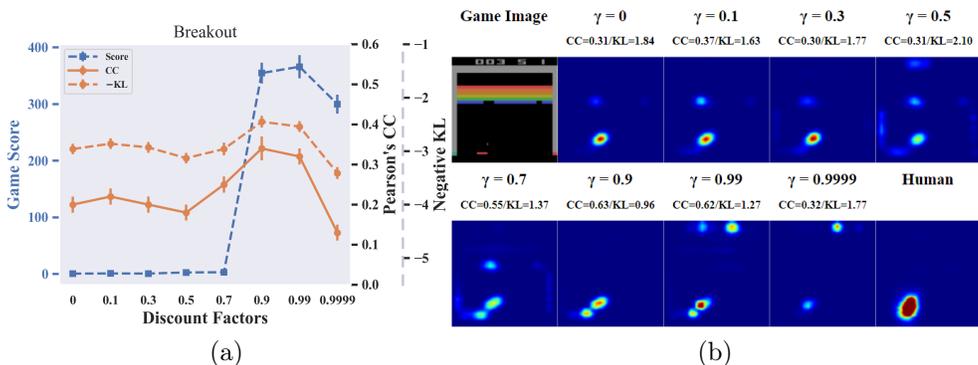


Figure A.10: Breakout: (a) The RL agent’s attention is most similar to human’s when  $\gamma = 0.9$ . (b) Human attention is on the paddle and the ball. Setting  $\gamma > 0.9$  makes the agent attend to the score at the top of the image.

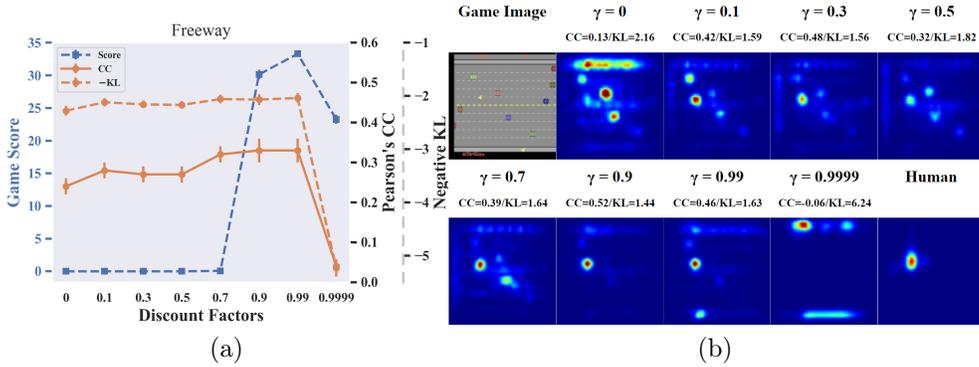


Figure A.11: Freeway: (a) The RL agent’s attention is most similar to human’s when  $\gamma = 0.9$  and  $\gamma = 0.99$ . (b) Human attention is on the yellow chicken being controlled to cross the highway.

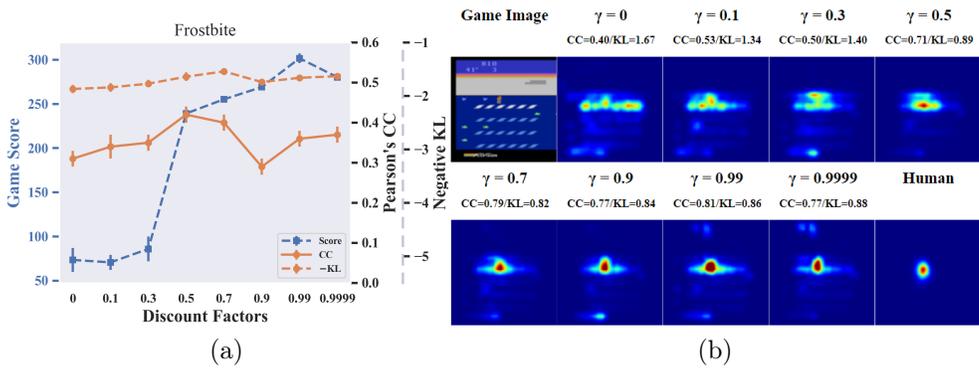


Figure A.12: Frostbite: (a) The RL agent’s attention is most similar to human’s when  $\gamma = 0.7$  (CC) or  $0.5$  (KL). (b) Human attention is on the little person being controlled in the middle.

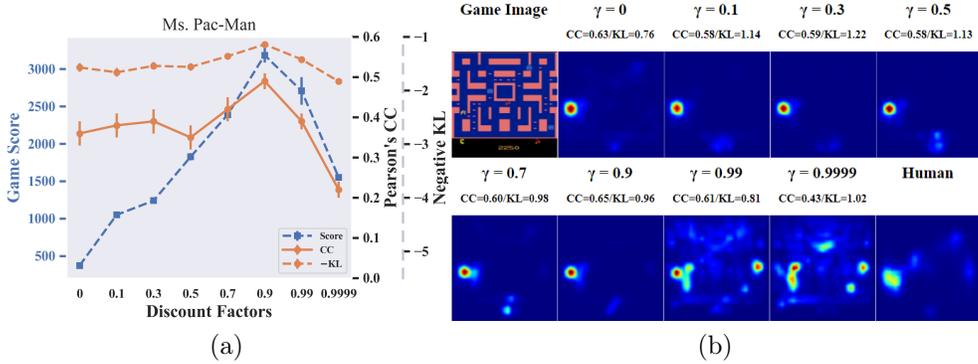


Figure A.13: Ms.Pac-Man: (a) The RL agent’s attention is most similar to human’s when  $\gamma = 0.9$ . Note that choosing this value and deviating from the default  $\gamma = 0.99$  lead to a better performance. (b) Human attention is mostly on the Pac-Man on the left side. Setting  $\gamma > 0.9$  distracts the agent to attend to other objects.

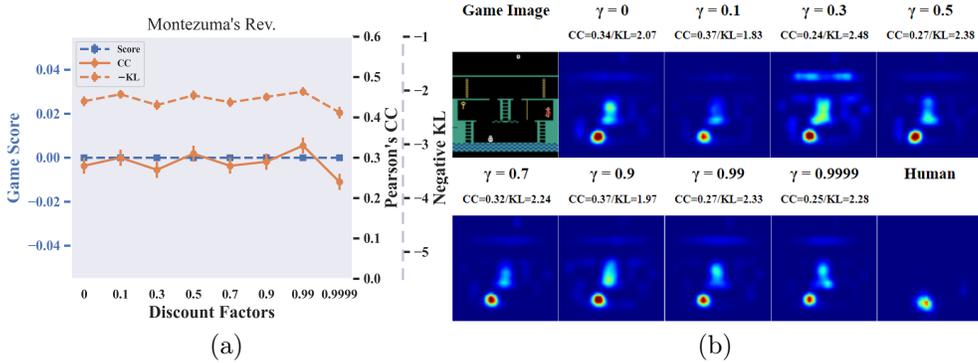


Figure A.14: Montezuma’s Revenge: (a) The RL agent’s attention is most similar to human’s when  $\gamma = 0.99$ . Note that this is a difficult game for RL agents and they never learn to score. (b) Human attention is on the enemy at the bottom.

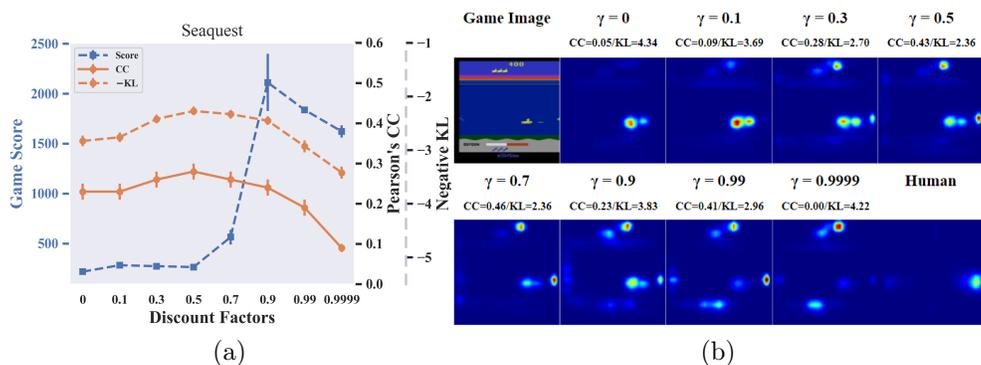


Figure A.15: Seaquest: (a) The RL agent’s attention is most similar to human’s when  $\gamma = 0.5$ . Note that choosing  $\gamma = 0.9$  and deviating from the default  $\gamma = 0.99$  lead to a better performance. (b) Human attention is on an appearing enemy on the right side. With  $\gamma > 0.9$  the RL agent also learns to attend to the oxygen bar at the bottom.

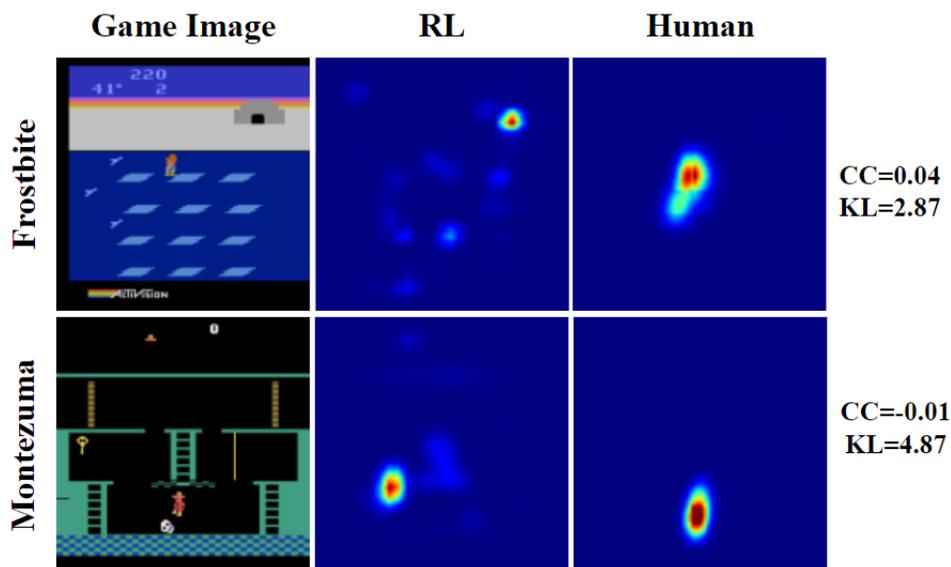


Figure A.16: Games in which human attention and RL agents' attention are more different in the failure states than the normal states. This indicates that in these games the mistakes are likely caused by wrong attention which subsequently led to wrong decisions. Frostbite: The RL agent is attending to the entrance of the Igloo. It should attend to the little person in the middle like humans do to avoid an incoming enemy from the left. Montezuma's Revenge: The RL agent is attending to the bottom of the ladder. It should attend the little person and the enemy to escape from the dangerous situation.

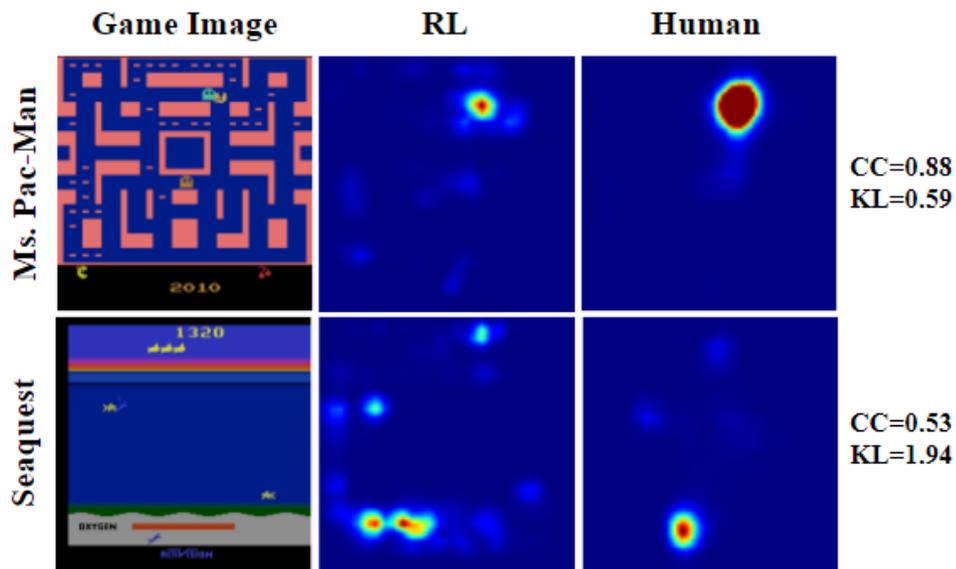


Figure A.17: Games in which human attention and RL agents' attention are more similar in the failure states than the normal states. This suggests that they generally agree on the objects to be attended to. But the RL agents made wrong decisions due to its suboptimal policy. Ms.Pac-Man: The agent and the human both attend to the Pac-Man which is about to be captured by the cyan enemy ghost. The agent failed to run away from it. Seaquest: The agent and the human both attend to the empty oxygen bar at the bottom. The agent failed to refill oxygen before it runs out.

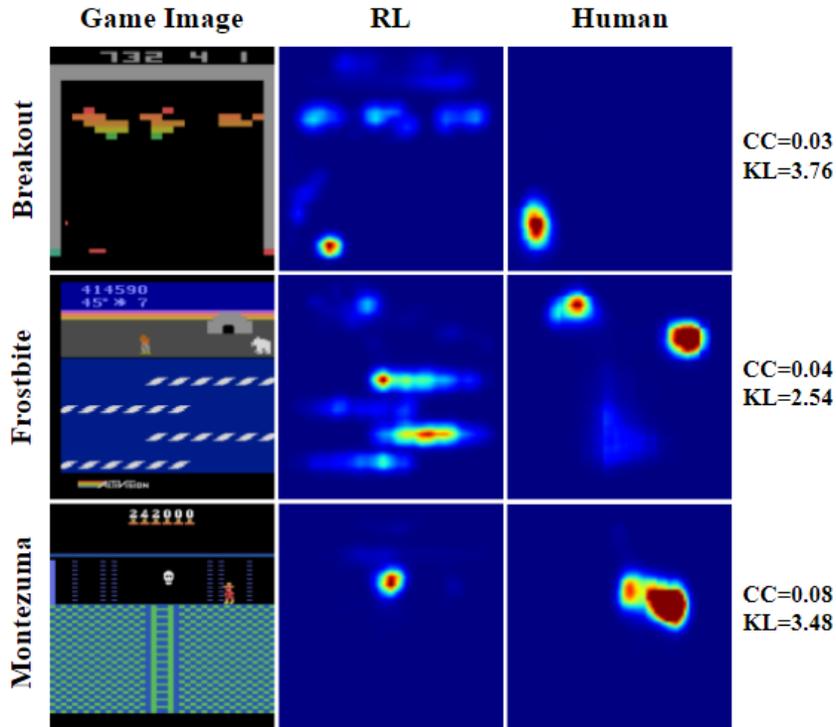


Figure A.18: Games in which human attention and RL agents' attention are more different in unseen states than the normal states. This is mostly due to new objects that the agents have never encountered. Breakout: The CC value drops significantly due to unseen spatial layouts of the bricks. The KL does not change much because there are no new objects so the agent can still attend to human attended objects like the ball on the left. Frostbite: Human attention is around the polar bear (a new object) at the upper right corner. Montezuma's Revenge: Human attention is on the fire beacon (a new object). The RL agent's attention is on the skull which is a familiar object.

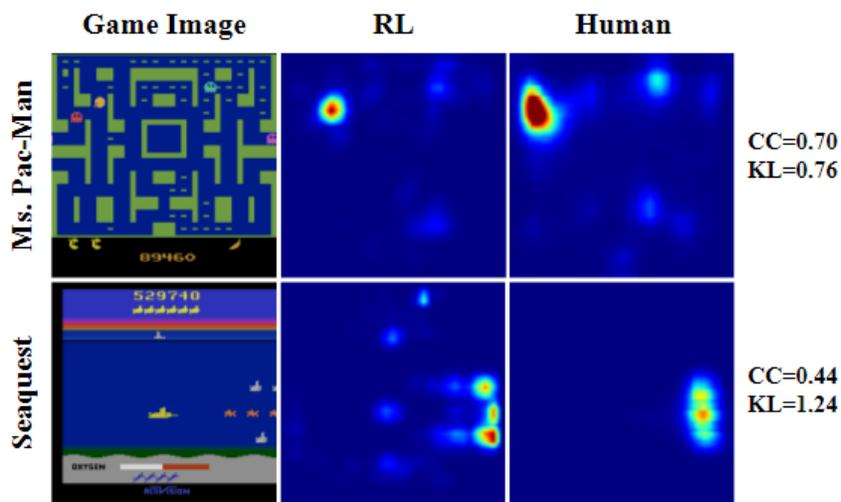


Figure A.19: Games in which human attention and RL agents' attention are more similar in the unseen states than the normal states. This is because there are no new objects in these unseen states – objects move much faster and appear in larger numbers. The player often encounters dangerous states that are close to failure. As shown in Appendix Fig. A.17 human attention and RL agent's attention are often similar in failure states for these two games. Ms. Pac-Man: The agent and the human both attend to the Pac-Man which is about to be captured by the red enemy ghost. Seaquest: The agent and the human both attend to the enemies on the right side.

## Appendix B

### The Modularization Hypothesis

#### B.1 Bayesian Inverse Reinforcement Learning

For comparison, a Bayesian IRL (BIRL) agent without modularization and assumes a fixed discount factor [244] is also implemented. Bayesian IRL leverages demonstrated state-action pairs, treats them individually as evidence for the underlying reward function, and therefore can express the likelihood of reward functions given demonstrations. The normalizing factor for computing the probability of reward functions is hard to compute, hence Bayesian IRL instead adopts a Monte Carlo Markov Chain (MCMC) sampling method to acquire a set of reward samples using the unnormalized likelihood function [244]. To compute the likelihood of a given reward function during sampling, it is required to compute the Q-values for all the state-action pairs in the demonstration set, which means solving a reinforcement learning (RL) problem given the Markov Decision Process (MDP). Therefore, Bayesian IRL is indeed a very computationally expensive algorithm.

To make our human experiment environment tractable by Bayesian IRL, the virtual room is discretized into a 2D gridworld of size  $32 \times 24$  with  $0.2 \times 0.2$   $m^2$  cells (Fig. B.1). Each cell is a state in the MDP. The actions are discretized

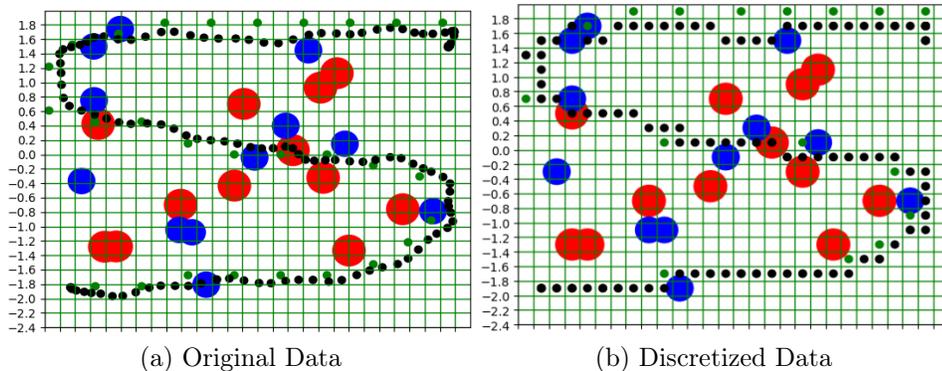


Figure B.1: Example of data discretization in the VR navigation experiment.

into 8 directions so that an agent can move to any adjacent state in the gridworld. The (center) location of targets, obstacles, and waypoints are treated as different feature points, which contribute to each state’s feature by distance. The problem is formulated as learning the weights for the three different features: targets, obstacles, and waypoints. The three features are represented using three different continuous values at each state. More specifically, the closer a state is to a target/obstacle/waypoint, the higher the feature value for the particular object at that state. The reward at any given state is computed as the linear combination of these features using their corresponding weights. The observations are a set of state-action pairs extracted from the human’s trajectory, which are fitted to the discretization of the space.

The parameters for Bayesian IRL are set empirically. The confidence factor  $\alpha$  is set at 80 and the chain length is set to be 3000 (since there are only three values, i.e. feature weights, to be tweaked, which is relatively small). A value of 0.5 is used as the discount factor for MDPs with the assumption that

the decision-making process of humans tends to prefer immediate rewards.

## Appendix C

### Neural Basis of Attention and Modularization

#### C.1 Cost of Population Coding

From a purely mathematical perspective, we here analyze the cost of basic rate coding models. We turn to the basic issue of estimating a scalar reliably from spikes that are generated with a probabilistic model. The model chosen is simpler than a Poisson model but is elegant. It illustrates the point that estimating a scalar accurately by counting spikes that are generated probabilistically takes a lot of them.

Let's use the Hoeffding's inequality [129] where  $m$  spikes are modeled as  $Z_1, \dots, Z_m$  from a Bernoulli distribution where  $P(Z_i = 1) = \phi$  and  $P(Z_i = 0) = 1 - \phi$ . The estimate of the mean is given by  $\hat{\phi} = \frac{1}{m} \sum_i Z_i$ . let any  $\mu \geq 0$  be a fixed error criterion. Then applying the inequality,

$$P(|\phi - \hat{\phi}| > \mu) \leq 2e^{-2\mu^2 m}.$$

This formula allows one to estimate the number of cells to be counted to bound the error of the estimate, which is done in Fig. C.1. This figure shows that the rate/population model may need thousands of neurons or spikes to signal a scalar value reliably (within a reasonable error bound). Meanwhile, the

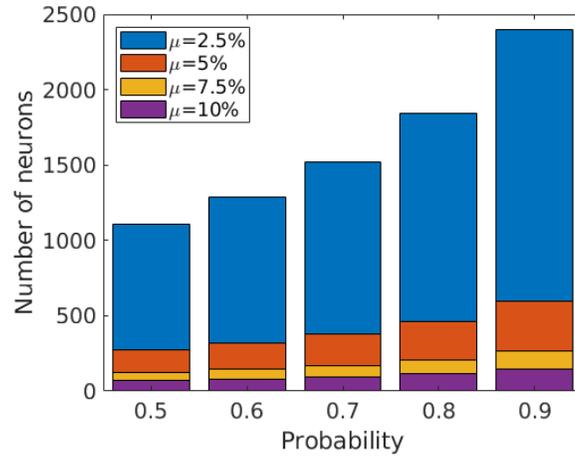


Figure C.1: Estimating the cost of population code that assumes rate code. Using a Bernoulli model for spike generation where the mean probability is set to a hypothetical “rate,” one can calculate how many spikes are needed in the estimate to bound the error in the estimate. For example, it takes at least 70 coding neurons, such that coding error will be within  $\mu = 10\%$  with probability at least 0.5 (first column, purple). It takes at least 2397 coding neurons, such that coding error will be within  $\mu = 2.5\%$  with a probability of at least 0.9 (last column, blue). Keep in mind these estimates are for just one scalar.

proposed gamma phase coding model only requires a single spike to represent this value.

## C.2 Sparse Coding and Neuron Selection

Instead of arbitrary computations, we use multiple instances of the familiar computation of representing small image patches with receptive fields in the striate cortex using sparse coding [225, 226]. The probabilistic parallel selection algorithm results in good coding quality with 50 to 100 neurons, as shown in Fig. C.2. Figure C.3 shows more coding examples when 8 image

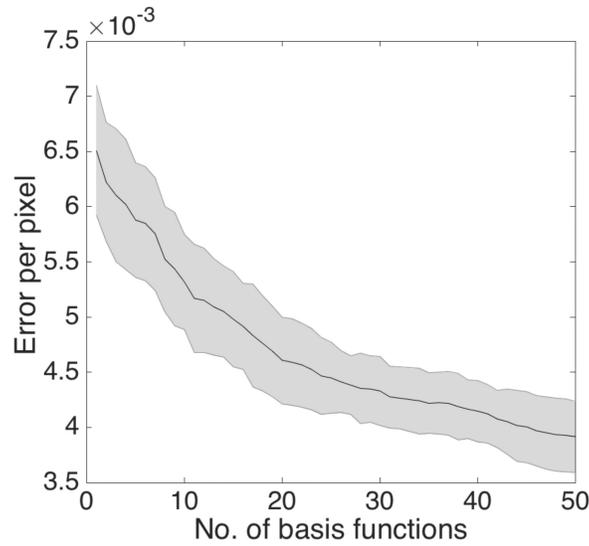
patches are encoded in parallel, using the proposed neuron selection algorithm. The main result here is that the sparse coding algorithm was implemented correctly.

### **C.3 A New Way of Interpreting Spike Data**

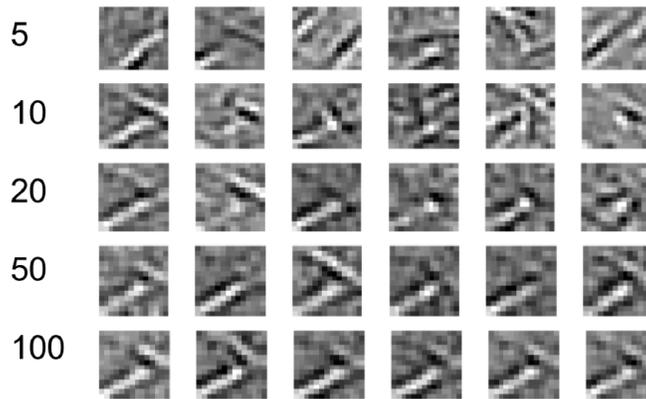
The proposed model would greatly change the way we interpret spike data. The spikes generated by the simulation are shown in Fig. C.4. The simulation uses 1000 coding neurons but for clarity in the visualization, a representative 50 cells are shown. Fig. C.4a illustrates the simulation's spikes as might be obtained with a conventional multi-cell recording. In Fig. C.4b, all spikes from cells that are coding the same image patch have a common color denoting their instantaneous modulation frequency. Owing to being generated in the context of specific gamma frequencies, individual spikes can be identified with the specific image patch that they are coding. Without the multiprocessing context, the spike trains appear very conventional, but once the generating context is available, the spikes can be readily associated with their processes. Additionally, it can be seen by inspection that a single coding cell's spikes can encode different patches.

### **C.4 Neural Recording Data**

One trial from a pyramidal cell is shown in Fig. C.5 with four spikes extracted from 7 seconds of a cell's membrane potential using patch-clamp data. The overlay clearly shows that the spikes are related to the rise in the



(a)



(b)

Figure C.2: The effect of the number of coding neurons on the accuracy of the parallel probabilistic coding algorithm. (a) Reconstruction mean squared error as a function of the number of coding neurons, shaded area indicates the standard error of the mean. (b) Sets of six independent reconstructions of the same image patch using increasing numbers of coding cells. We show successive reconstructions using 5, 10, 20, 50 and 100 coding cells using the parallel probabilistic selection algorithm. If 50 to 100 neurons are used the resultant code becomes accurate.

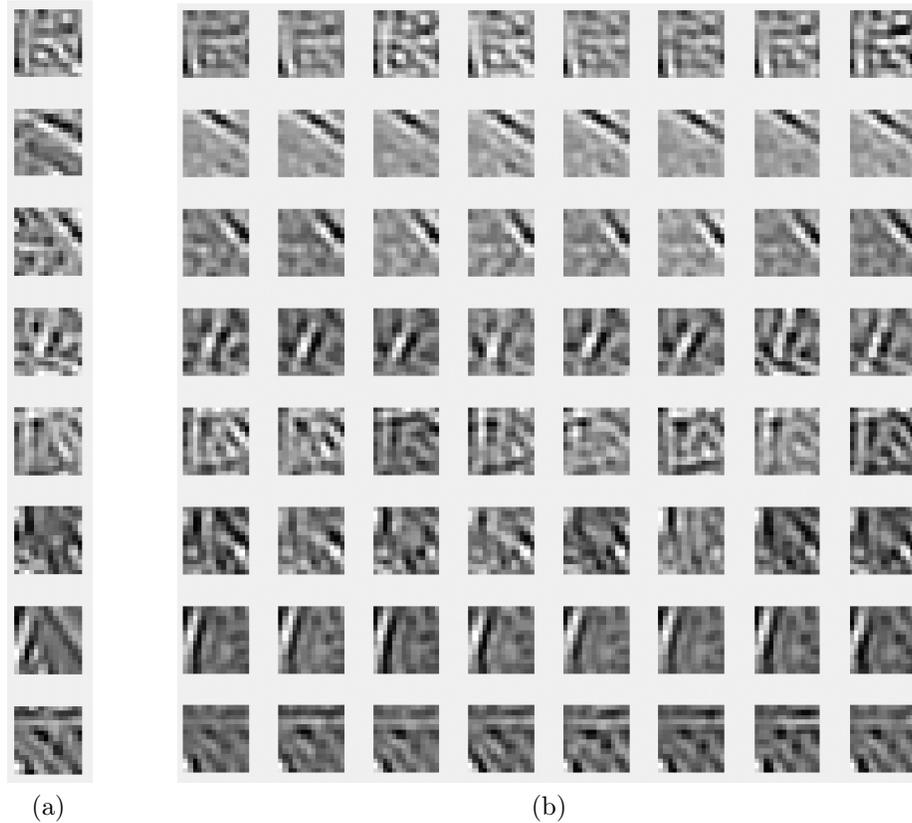
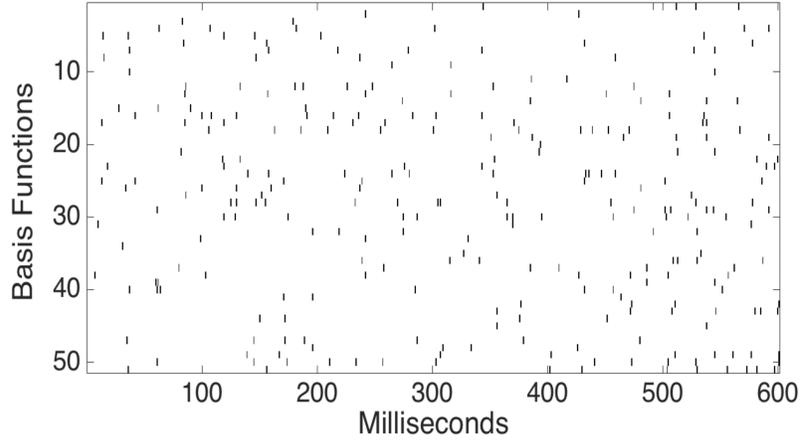
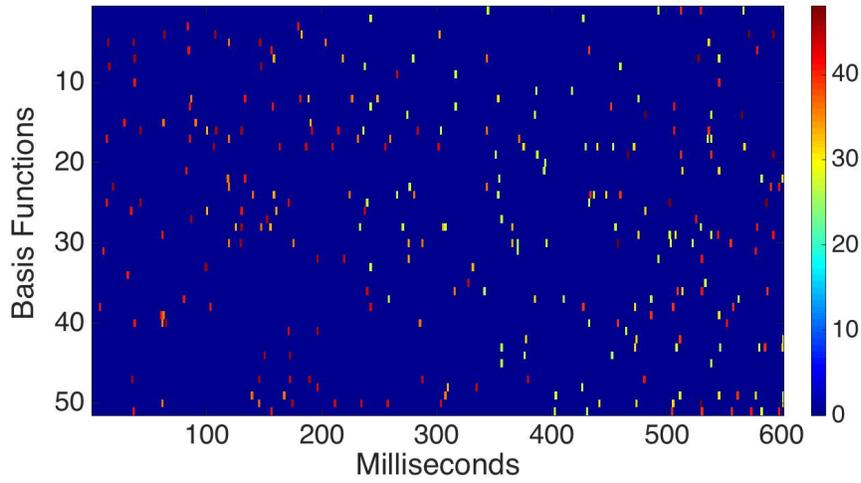


Figure C.3: Given an image patch, coding neurons are selected probabilistically. (a) Eight image patches to be coded in the leftmost column simultaneously, representing the eight ongoing neural processes. (b) Each of these is reconstructed eight different times using 50 coding cells. The consequence of over-completeness and probabilistic selection is that the image patch can be coded using very distinct sets of neurons. The reconstructions are accurate with a slight tendency to regularize the patterns.



(a)

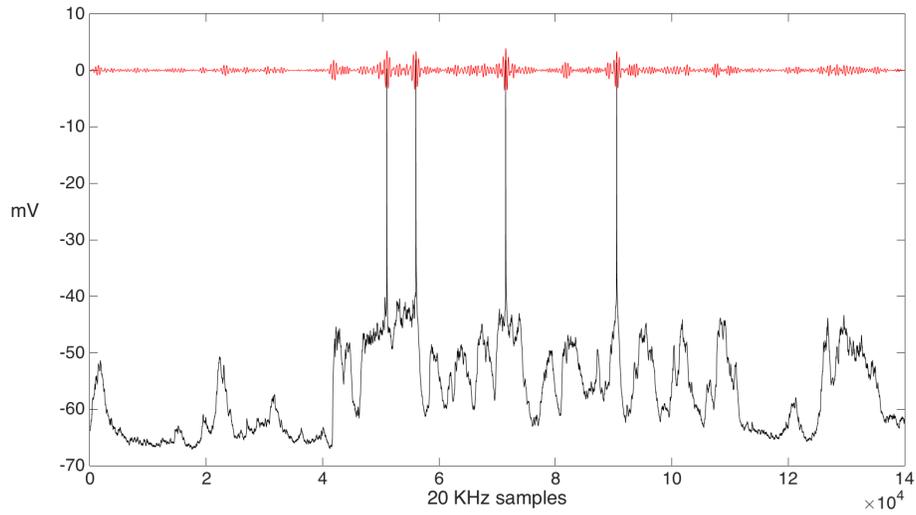


(b)

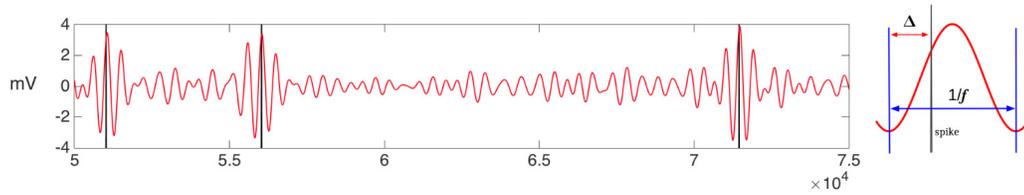
Figure C.4: GSM provides a new way of interpreting cell recording data. Spikes are rendered with 2 ms thickness for enhanced visibility. (a) Without the gamma frequency context, spikes can be difficult to interpret. (b) For the same data, we denote each process with a unique color indicating its dedicated gamma frequency. Each spike is then associated with the process that encodes a component of a particular image patch.

gamma potential. Each spike codes an analog quantity that is signaled by the delay  $\Delta$  between the time of the gamma wavelet trough and the spike time. For the first three spikes, the delays are 3, 4, and 6 milliseconds. One important observation is that the gamma frequency on a cell's soma can be instantaneously modulated. For identifying the frequency,  $f$  is measured from the separation distance of the zero crossings, resulting in frequencies of 43, 46, and 48 Hz.

The putative powerful effect of the small gamma amplitudes can challenge one's intuition, but one needs to keep in mind the neck of an action potential is exquisitely regulated by the cells in our study to be within a millivolt of negative 37 millivolts, a much smaller excursion compared to the  $\pm 4$  millivolts observed in the filtered gamma amplitude signal.



(a)



(b)

Figure C.5: (a) An example patch-clamp data from a single trial. The filtered somatic potential in the gamma frequency range with passband interval  $[30, 80]$  Hz shows that spikes are in the first quadrant of the gamma cycle. (b) The first three spikes that are labeled by their associated gamma frequencies. The frequencies and delays are  $\{f, \Delta\} = \{(43\text{Hz}, 3\text{ms}), (46\text{Hz}, 4\text{ms}), (48\text{Hz}, 6\text{ms})\}$ . [Results derived from patch clamp data as a courtesy of the Luc Gentet lab at the Lyon Neuroscience Research Center].

## Bibliography

- [1] Pieter Abbeel, Adam Coates, and Andrew Y Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.
- [2] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.
- [3] David Abel, Dilip Arumugam, Kavosh Asadi, Yuu Jinnai, Michael L Littman, and Lawson LS Wong. State abstraction as compression in apprenticeship learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2019.
- [4] David Abel, John Salvatier, Andreas Stuhlmüller, and Owain Evans. Agent-agnostic human-in-the-loop reinforcement learning. *NeurIPS Workshop on the Future of Interactive Learning Machines*, 2017.
- [5] Moshe Abeles. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991.
- [6] Alnour Alharin, Thanh-Nam Doan, and Mina Sartipi. Reinforcement learning interpretation methods: A survey. *IEEE Access*, 8:171058–171077, 2020.

- [7] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. 2016.
- [8] Riku Arakawa, Sosuke Kobayashi, Yuya Unno, Yuta Tsuboi, and Shinichi Maeda. Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback. *arXiv preprint arXiv:1810.11748*, 2018.
- [9] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.
- [10] Ronald C Arkin. Motor schema—based mobile robot navigation. *The International journal of robotics research*, 8(4):92–112, 1989.
- [11] Reuben M Aronson, Thiago Santini, Thomas C Kübler, Enkelejda Kasneci, Siddhartha Srinivasa, and Henny Admoni. Eye-hand behavior in human-robot shared manipulation. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 4–13, 2018.
- [12] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *arXiv preprint arXiv:1806.06877*, 2018.
- [13] Bassam V Atallah and Massimo Scanziani. Instantaneous modulation of gamma oscillation frequency by balancing excitation with inhibition.

*Neuron*, 62(4):566–577, 2009.

- [14] Monica Babes, Vukosi Marivate, Kaushik Subramanian, and Michael L Littman. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 897–904, 2011.
- [15] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*, pages 507–517. PMLR, 2020.
- [16] Michael Bain and Claude Sommut. A framework for behavioural cloning. *Machine intelligence*, 15(15):103, 1999.
- [17] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.
- [18] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. Goal inference as inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29, 2007.
- [19] Dana Ballard and Janneke Jehee. Dual roles for spike signaling in cortical neural populations. *Frontiers in computational neuroscience*, 5:22, 2011.
- [20] Dana Ballard and Ruohan Zhang. Cortical spike multiplexing using gamma frequency latencies. *bioRxiv*, page 313320, 2018.

- [21] Dana Ballard, Ruohan Zhang, and Luc Gentet. Cortical spikes use analog sparse coding. *bioRxiv*, 2020.
- [22] Dana H Ballard. *Brain Computation as Hierarchical Abstraction*. MIT Press, 2015.
- [23] Dana H Ballard, Dmitry Kit, Constantin A Rothkopf, and Brian Sullivan. A hierarchical modular architecture for embodied cognition. *Multisensory research*, 26(1-2):177–204, 2013.
- [24] Dana H Ballard and Ruohan Zhang. The hierarchical evolution in human vision modeling. *Topics in Cognitive Science*, 2020.
- [25] Albert Bandura, Dorothea Ross, and Sheila A Ross. Transmission of aggression through imitation of aggressive models. *The Journal of Abnormal and Social Psychology*, 63(3):575, 1961.
- [26] Horace B Barlow. Single units and sensation: a neuron doctrine for perceptual psychology? *Perception*, 1(4):371–394, 1972.
- [27] Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.
- [28] Andre M Bastos, Julien Vezoli, and Pascal Fries. Communication through coherence with inter-areal delays. *Current opinion in neurobiology*, 31:173–180, 2015.

- [29] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458. PMLR, 2017.
- [30] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 2012.
- [31] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.
- [32] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.
- [33] Toby Berger. Rate-distortion theory. *Wiley Encyclopedia of Telecommunications*, 2003.
- [34] Michael J Berry II and Markus Meister. Refractoriness and neural precision. In *Advances in Neural Information Processing Systems*, pages 110–116, 1998.
- [35] Erdem Bıyık, Dylan P Losey, Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *arXiv preprint arXiv:2006.14091*, 2020.

- [36] Rafal Bogacz, Eduardo Martin Moraud, Azzedine Abdi, Peter J Magill, and Jérôme Baufreton. Properties of neurons in external globus pallidus can support optimal action selection. *PLoS Comput Biol*, 12(7):e1005004, 2016.
- [37] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [38] Lyle J Borg-Graham, Cyril Monier, and Yves Fregnac. Visual input evokes transient and strong shunting inhibition in visual cortical neurons. *Nature*, 393(6683):369–373, 1998.
- [39] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE transactions on image processing*, 24(12):5706–5722, 2015.
- [40] Ali Borji and Laurent Itti. Defending yabus: Eye movements reveal observers’ task. *Journal of Vision*, 14(3):29–29, 2014.
- [41] Ali Borji and Laurent Itti. Cat2000: A large scale fixation dataset for boosting saliency research. *arXiv preprint arXiv:1505.03581*, 2015.
- [42] Geoffrey M Boynton. A framework for describing the effects of attention on visual responses. *Vision research*, 49(10):1129–1143, 2009.

- [43] Vincent Bringuier, Frederic Chavane, Larry Glaeser, and Yves Frégnac. Horizontal propagation of visual activity in the synaptic integration field of area 17 neurons. *Science*, 283(5402):695–699, 1999.
- [44] Daniel S Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. *arXiv preprint arXiv:1904.06387*, 2019.
- [45] Nicolas Brunel and Xiao-Jing Wang. What determines the frequency of fast network oscillations with irregular neural discharges? i. synaptic dynamics and excitation-inhibition balance. *Journal of neurophysiology*, 90(1):415–430, 2003.
- [46] Nicolas Brunet, Conrado A Bosman, Mark Roberts, Robert Oostenveld, Thilo Womelsdorf, Peter De Weerd, and Pascal Fries. Visual cortical gamma-band activity during free viewing of natural images. *Cerebral cortex*, 25(4):918–926, 2013.
- [47] Samuel P Burns, Dajun Xing, and Robert M Shapley. Is gamma-band activity in the local field potential of v1 cortex a “clock” or filtered noise? *Journal of Neuroscience*, 31(26):9658–9664, 2011.
- [48] György Buzsáki and Xiao-Jing Wang. Mechanisms of gamma oscillations. *Annual review of neuroscience*, 35:203–225, 2012.
- [49] Zoya Bylinskii, Phillip Isola, Constance Bainbridge, Antonio Torralba, and Aude Oliva. Intrinsic and extrinsic effects on image memorability.

*Vision research*, 116:165–178, 2015.

- [50] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark.
- [51] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark, 2015.
- [52] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3):740–757, 2018.
- [53] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):740–757, 2019.
- [54] Zoya Bylinskii, Adrià Recasens, Ali Borji, Aude Oliva, Antonio Torralba, and Frédo Durand. Where should saliency models look next? In *European Conference on Computer Vision*, pages 809–824. Springer, 2016.
- [55] Rudolf N Cardinal. Neural systems implicated in delayed and probabilistic reinforcement. *Neural Networks*, 19(8):1277–1301, 2006.

- [56] Luis Carrillo-Reid, Shuting Han, Ekaterina Taralova, Tony Jebara, and Rafael Yuste. Identification and targeting of cortical ensembles. *bioRxiv*, page 226514, 2017.
- [57] Pablo Samuel Castro, Subhdeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A Research Framework for Deep Reinforcement Learning. 2018.
- [58] Thomas Cederborg, Ishaan Grover, Charles L Isbell, and Andrea L Thomaz. Policy shaping with human teachers. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3366–3372. AAAI Press, 2015.
- [59] Matthew Chalk, Jose L Herrero, Mark A Gieselmann, Louise S Delicato, Sascha Gotthardt, and Alexander Thiele. Attention reduces stimulus-driven gamma frequency oscillations and spike field coherence in v1. *Neuron*, 66(1):114–125, 2010.
- [60] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.
- [61] Yuying Chen, Congcong Liu, Bertram E Shi, and Ming Liu. Robot navigation in crowds by graph convolutional networks with attention

- learned from human gaze. *IEEE Robotics and Automation Letters*, 5(2):2754–2761, 2020.
- [62] Yuying Chen, Congcong Liu, Lei Tai, Ming Liu, and Bertram E Shi. Gaze training by modulated dropout improves imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7756–7761. IEEE, 2019.
- [63] Jaedeug Choi and Kee-Eung Kim. Hierarchical bayesian inverse reinforcement learning. *IEEE transactions on cybernetics*, 45(4):793–805, 2015.
- [64] Mark M Churchland, M Yu Byron, John P Cunningham, Leo P Sugrue, Marlene R Cohen, Greg S Corrado, William T Newsome, Andrew M Clark, Paymon Hosseini, Benjamin B Scott, et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, 13(3):369, 2010.
- [65] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.
- [66] Paul Cisek. Making decisions through a distributed consensus. *Current opinion in neurobiology*, 22(6):927–936, 2012.

- [67] Logan Cross, Jeff Cockburn, Yisong Yue, and John P O’Doherty. Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments. *Neuron*, 2020.
- [68] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976, 2017.
- [69] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100, 2017.
- [70] Nathaniel D Daw, Samuel J Gershman, Ben Seymour, Peter Dayan, and Raymond J Dolan. Model-based influences on humans’ choices and striatal prediction errors. *Neuron*, 69(6):1204–1215, 2011.
- [71] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience*, volume 806. Cambridge, MA: MIT Press, 2001.
- [72] Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In *Advances in neural information processing systems*, pages 271–278, 1993.
- [73] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, pages 11698–11709, 2019.

- [74] Gabriel V de la Cruz, Yunshu Du, and Matthew E Taylor. Pre-training with non-expert human demonstration for deep reinforcement learning. *arXiv preprint arXiv:1812.08904*, 2018.
- [75] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [76] Tao Deng, Hongmei Yan, Long Qin, Thuyen Ngo, and BS Manjunath. How do drivers allocate their potential attention? driving fixation prediction via convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):2146–2154, 2019.
- [77] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. <https://github.com/openai/baselines>, 2017.
- [78] Gabriel Diaz, Joseph Cooper, Constantin Rothkopf, and Mary Hayhoe. Saccades to future ball location reveal memory-based prediction in a virtual-reality interception task. *Journal of vision*, 13(1):20–20, 2013.
- [79] Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res.(JAIR)*, 13:227–303, 2000.

- [80] Kenji Doya. Modulators of decision making. *Nature neuroscience*, 11(4):410–416, 2008.
- [81] Shaul Druckmann and Dmitri B Chklovskii. Over-complete representations on recurrent neural networks can support persistent percepts. In *Advances in Neural Information Processing Systems*, pages 541–549, 2010.
- [82] Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Thomas L Griffiths, and Alexei A Efros. Investigating human priors for playing video games. *arXiv preprint arXiv:1802.10217*, 2018.
- [83] Michael Eickenberg, Alexandre Gramfort, Gaël Varoquaux, and Bertrand Thirion. Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage*, 152:184–194, 2017.
- [84] Colin G Evans, Jian Jing, Steven C Rosen, and Elizabeth C Cropper. Regulation of spike initiation and propagation in anaplysia sensory neuron: Gating-in via central depolarization. *Journal of Neuroscience*, 23(7):2920–2931, 2003.
- [85] Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, pages 1–8, 2019.
- [86] Rui Fang, Malcolm Doering, and Joyce Y Chai. Embodied collaborative referring expression generation in situated human-robot interaction. In

- Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 271–278, 2015.
- [87] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003.
- [88] William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.
- [89] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *ICML*, pages 49–58, 2016.
- [90] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.
- [91] DJ Foster, RGM Morris, Peter Dayan, et al. A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16, 2000.
- [92] Julien Fournier, Aman B Saleem, E Mika Diamanti, Miles J Wells, Kenneth D Harris, and Matteo Carandini. Modulation of visual cortex by hippocampal signals. *bioRxiv*, page 586917, 2019.
- [93] Pascal Fries. Rhythms for cognition: communication through coherence. *Neuron*, 88(1):220–235, 2015.

- [94] Pascal Fries, Danko Nikolić, and Wolf Singer. The gamma cycle. *Trends in neurosciences*, 30(7):309–316, 2007.
- [95] Pascal Fries, John H Reynolds, Alan E Rorie, and Robert Desimone. Modulation of oscillatory neuronal synchronization by selective visual attention. *Science*, 291(5508):1560–1563, 2001.
- [96] Samuel J Gershman, Bijan Pesaran, and Nathaniel D Daw. Human reinforcement learning subdivides structured action spaces by learning effector-specific values. *The Journal of Neuroscience*, 29(43):13524–13531, 2009.
- [97] George L Gerstein and Benoit Mandelbrot. Random walk models for the spike activity of a single neuron. *Biophysical journal*, 4(1):41–68, 1964.
- [98] Wulfram Gerstner, Andreas K Kreiter, Henry Markram, and Andreas VM Herz. Neural codes: firing rates and beyond. *Proceedings of the National Academy of Sciences*, 94(24):12740–12741, 1997.
- [99] Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5):350, 2013.
- [100] Thomas Gisiger and Mounir Boukadoum. Mechanisms gating the flow of information in the cortex: what they might look like and what their uses may be. *Frontiers in computational neuroscience*, 5:1, 2011.
- [101] Alessandro Giusti, Jérôme Guzzi, Dan C Cireşan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen

- Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2016.
- [102] Vinicius G Goecks, Gregory M Gremillion, Vernon J Lawhern, John Valasek, and Nicholas R Waytowich. Efficiently combining human demonstrations and interventions for safe training of autonomous systems in real-time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2462–2470, 2019.
- [103] Tim Gollisch and Markus Meister. Rapid neural coding in the retina with relative spike latencies. *science*, 319(5866):1108–1111, 2008.
- [104] Usha Goswami. *Cognitive development: The learning brain*. Psychology Press, 2008.
- [105] Jacqueline Gottlieb, Mary Hayhoe, Okihide Hikosaka, and Antonio Rangel. Attention, reward, and information seeking. *Journal of Neuroscience*, 34(46):15497–15504, 2014.
- [106] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International Conference on Machine Learning*, pages 1792–1801, 2018.
- [107] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International Conference on Machine Learning*, pages 1792–1801, 2018.

- [108] Christopher Grimm, Dilip Arumugam, Siddharth Karamcheti, David Abel, Lawson LS Wong, and Michael L Littman. Modeling latent attention within neural networks. *arXiv preprint arXiv:1706.00536*, 2017.
- [109] Amiram Grinvald, Edmund E Lieke, Ron D Frostig, and Rina Hildesheim. Cortical point-spread function and long-range lateral interactions revealed by real-time optical imaging of macaque monkey primary visual cortex. *Journal of Neuroscience*, 14(5):2545–2568, 1994.
- [110] Lin Guan, Mudit Verma, Sihang Guo, Ruohan Zhang, and Subbarao Kambhampati. Explanation augmented feedback in human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2006.14804*, 2020.
- [111] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, pages 399–468, 2003.
- [112] Sihang Guo, Bharath Masetty, Ruohan Zhang, Dana Ballard, and Mary Hayhoe. Modeling human multitasking behavior in video games through modular reinforcement learning. *Journal of Vision*, 20(11):1552–1552, 2020.
- [113] Piyush Gupta, Nikaash Puri, Sukriti Verma, Sameer Singh, Dhruv Kayastha, Shripad Deshmukh, and Balaji Krishnamurthy. Explain your move: Understanding agent actions using focused feature saliency. *arXiv preprint arXiv:1912.12191*, 2019.

- [114] William H Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint arXiv:1907.13440*, 2019.
- [115] Masahiko Haruno, Tomoe Kuroda, Kenji Doya, Keisuke Toyama, Minoru Kimura, Kazuyuki Samejima, Hiroshi Imamizu, and Mitsuo Kawato. A neural correlate of reward-based behavioral learning in caudate nucleus: a functional magnetic resonance imaging study of a stochastic decision task. *The Journal of Neuroscience*, 24(7):1660–1665, 2004.
- [116] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in cognitive sciences*, 9(4):188–194, 2005.
- [117] Mary Hayhoe and Dana Ballard. Modeling task control of eye movements. *Current Biology*, 24(13):R622–R628, 2014.
- [118] Mary M Hayhoe. Vision and action. *Annual review of vision science*, 3:389–413, 2017.
- [119] Sen He, Hamed R Tavakoli, Ali Borji, Yang Mi, and Nicolas Pugeault. Understanding and visualizing deep visual saliency models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10206–10215, 2019.
- [120] Sen He, Hamed R Tavakoli, Ali Borji, and Nicolas Pugeault. Human attention in image captioning: Dataset and analysis. In *Proceedings of*

*the IEEE International Conference on Computer Vision*, pages 8529–8538, 2019.

- [121] Jens Herberholz, Brian L Antonsen, and Donald H Edwards. A lateral excitatory network in the escape circuit of crayfish. *Journal of Neuroscience*, 22(20):9078–9085, 2002.
- [122] John A Hertz, Anders S Krogh, and Richard G Palmer. *Introduction to the theory of neural computation*, volume 1. Basic Books, 1991.
- [123] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, 2017.
- [124] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [125] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

- [126] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, page 106685, 2020.
- [127] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. <https://github.com/hill-a/stable-baselines>, 2018.
- [128] Elena Hitzel, Matthew Tong, Alexander Schütz, and Mary Hayhoe. Objects in the peripheral visual field influence gaze location in natural vision. *Journal of vision*, 15(12):e783–e783, 2015.
- [129] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.
- [130] Clay B Holroyd and Michael GH Coles. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological review*, 109(4):679, 2002.
- [131] John J Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535):33–36, 1995.
- [132] John J Hopfield and David W Tank. Computing with neural circuits: A model. *Science*, 233(4764):625–633, 1986.

- [133] Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *2016 11th ACM/IEEE international conference on human-robot interaction*, pages 83–90. IEEE, 2016.
- [134] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Tabletgaze dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, 28(5-6):445–461, 2017.
- [135] Wesley H Huang, Brett R Fajen, Jonathan R Fink, and William H Warren. Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, 54(4):288–299, 2006.
- [136] Yifei Huang, Minjie Cai, Zhenqiang Li, Feng Lu, and Yoichi Sato. Mutual context network for jointly estimating egocentric gaze and action. *IEEE Transactions on Image Processing*, 29:7795–7806, 2020.
- [137] Joel Huber, John W Payne, and Christopher Puto. Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of consumer research*, 9(1):90–98, 1982.
- [138] Tobias Huber, Benedikt Limmer, and Elisabeth André. Benchmarking perturbation-based saliency maps for explaining deep reinforcement learning agents. *arXiv preprint arXiv:2101.07312*, 2021.

- [139] Mark Humphrys. Action selection methods using reinforcement learning. *From Animals to Animats*, 4:135–144, 1996.
- [140] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):21, 2017.
- [141] Alexander G Huth, Wendy A De Heer, Thomas L Griffiths, Frédéric E Theunissen, and Jack L Gallant. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532(7600):453–458, 2016.
- [142] Stephen Hutt, Caitlin Mills, Shelby White, Patrick J Donnelly, and Sidney K D’Mello. The eyes have it: Gaze-based detection of mind wandering during learning with an intelligent tutoring system. *International Educational Data Mining Society*, 2016.
- [143] Kei M Igarashi, Li Lu, Laura L Colgin, May-Britt Moser, and Edvard I Moser. Coordination of entorhinal–hippocampal ensemble activity during associative learning. *Nature*, 510(7503):143–147, 2014.
- [144] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.
- [145] Andrei I Ivanov and Ronald L Calabrese. Modulation of spike-mediated synaptic transmission by presynaptic background  $ca^{2+}$  in leech heart interneurons. *Journal of Neuroscience*, 23(4):1206–1218, 2003.

- [146] Rahul Iyer, Yuezhong Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 144–150, 2018.
- [147] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.
- [148] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.
- [149] Shailee Jain and Alexander Huth. Incorporating context into language encoding models for fmri. In *Advances in Neural Information Processing Systems*, pages 6628–6637, 2018.
- [150] Natasha Jaques, Cristina Conati, Jason M Harley, and Roger Azevedo. Predicting affect from gaze data during interaction with an intelligent tutoring system. In *International Conference on Intelligent Tutoring Systems*, pages 29–38. Springer, 2014.
- [151] Theo Jaunet, Romain Vuillemot, and Christian Wolf. Drlviz: Understanding decisions and memory in deep reinforcement learning. *arXiv*

*preprint arXiv:1909.02982*, 2019.

- [152] Janneke FM Jehee, Constantin Rothkopf, Jeffrey M Beck, and Dana H Ballard. Learning receptive fields using predictive feedback. *Journal of Physiology-Paris*, 100(1-3):125–132, 2006.
- [153] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080, 2015.
- [154] Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1181–1189, 2015.
- [155] Leif Johnson, Brian Sullivan, Mary Hayhoe, and Dana Ballard. Predicting human visuomotor behaviour in a driving task. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 369(1636):20130044, 2014.
- [156] Ho-Taek Joo and Kyung-Joong Kim. Visualization of deep reinforcement learning using grad-cam: How ai plays atari games? In *2019 IEEE Conference on Games (CoG)*, pages 1–2. IEEE, 2019.
- [157] Jelena Jovancevic, Brian Sullivan, and Mary Hayhoe. Control of attention and gaze in complex environments. *Journal of Vision*, 6(12):9–9, 2006.

- [158] Jelena Jovancevic-Misic and Mary Hayhoe. Adaptive gaze control in natural environments. *Journal of Neuroscience*, 29(19):6234–6238, 2009.
- [159] Sebastian Junges, Nils Jansen, Joost-Pieter Katoen, Ufuk Topcu, Ruohan Zhang, and Mary Hayhoe. Model checking for safe navigation among humans. In *International Conference on Quantitative Evaluation of Systems*, pages 207–222. Springer, 2018.
- [160] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
- [161] Paul S Katz. Synaptic gating: the potential to open closed doors. *Current Biology*, 13(14):R554–R556, 2003.
- [162] Mitsuo Kawato and Kazuyuki Samejima. Efficient reinforcement learning: computational theories, neuroscience and robotics. *Current opinion in neurobiology*, 17(2):205–212, 2007.
- [163] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.
- [164] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM, 2009.

- [165] W Bradley Knox and Peter Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 5–12. International Foundation for Autonomous Agents and Multiagent Systems, 2010.
- [166] W Bradley Knox and Peter Stone. Reinforcement learning from simultaneous human and mdp reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 475–482. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [167] Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Ranulfo Romo, Xue-Lian Qi, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *Elife*, 5:e10989, 2016.
- [168] Kilian Koepsell, Xin Wang, Vishal Vaingankar, Yichun Wei, Qingbo Wang, Daniel L Rathbun, Martin W Usrey, Judith Hirsch, and Friedrich T Sommer. Retinal oscillations carry visual information to cortex. *Frontiers in systems neuroscience*, 3:4, 2009.
- [169] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

- [170] Vitaly Kurin, Sebastian Nowozin, Katja Hofmann, Lucas Beyer, and Bastian Leibe. The atari grand challenge dataset. *arXiv preprint arXiv:1705.10998*, 2017.
- [171] Qiuxia Lai, Wenguan Wang, Salman Khan, Jianbing Shen, Hanqiu Sun, and Ling Shao. Human\textit {vs} machine attention in neural networks: A comparative study. *arXiv preprint arXiv:1906.08764*, 2019.
- [172] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.
- [173] Michael F Land. Vision, eye movements, and natural behavior. *Visual neuroscience*, 26(1):51–62, 2009.
- [174] Ayelet Nina Landau and Pascal Fries. Attention samples stimuli rhythmically. *Current biology*, 22(11):1000–1004, 2012.
- [175] Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudik, Yisong Yue, and Hal Daumé. Hierarchical imitation and reinforcement learning. In *International Conference on Machine Learning*, pages 2923–2932, 2018.
- [176] Olivier Le Meur and Thierry Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.
- [177] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

- [178] Daeyeol Lee, Hyojung Seo, and Min Whan Jung. Neural basis of reinforcement learning and decision making. *Annual review of neuroscience*, 35:287, 2012.
- [179] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.
- [180] Peter Lennie. The cost of cortical computation. *Current biology*, 13(6):493–497, 2003.
- [181] Yuan Chang Leong, Angela Radulescu, Reka Daniel, Vivian DeWoskin, and Yael Niv. Dynamic interaction between reinforcement learning and attention in multidimensional environments. *Neuron*, 93(2):451–463, 2017.
- [182] RA Lerch and CR Sims. Rate-distortion theory and computationally rational reinforcement learning. *Proceedings of Reinforcement Learning and Decision Making (RLDM) 2019*, pages 7–10, 2019.
- [183] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.
- [184] Dino J Levy and Paul W Glimcher. The root of all value: a neural common currency for choice. *Current opinion in neurobiology*, 22(6):1027–1038, 2012.

- [185] Jonathan Levy, Harold Pashler, and Erwin Boer. Central interference in driving: Is there any stopping the psychological refractory period? *Psychological science*, 17(3):228–235, 2006.
- [186] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.
- [187] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 619–635, 2018.
- [188] John Lisman. The theta/gamma discrete phase code occurring during the hippocampal phase precession may be a more general brain coding scheme. *Hippocampus*, 15(7):913–922, 2005.
- [189] John E Lisman and Ole Jensen. The theta-gamma neural code. *Neuron*, 77(6):1002–1016, 2013.
- [190] Alfred Lit. The magnitude of the pulfrich stereophenomenon as a function of target velocity. *Journal of Experimental Psychology*, 59(3):165, 1960.
- [191] Congcong Liu, Yuying Chen, Lei Tai, Haoyang Ye, Ming Liu, and Bertram E Shi. A gaze model improves autonomous driving. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, page 33. ACM, 2019.

- [192] Rodolfo R Llinas, Anthony A Grace, and Yosef Yarom. In vitro neurons in mammalian cortical layer 4 exhibit intrinsic oscillatory activity in the 10-to 50-hz frequency range. *Proceedings of the National Academy of Sciences*, 88(3):897–901, 1991.
- [193] Gordon D Logan. Executive control of thought and action. *Acta psychologica*, 60(2-3):193–210, 1985.
- [194] M. Lopes, F. Melo, and L. Montesano. Active learning for reward estimation in inverse reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, pages 31–46, 2009.
- [195] Steven J Luck and Edward K Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, 390(6657):279, 1997.
- [196] James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. Interactive learning from policy-dependent human feedback. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2285–2294. JMLR. org, 2017.
- [197] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.
- [198] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay,

- et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. *arXiv preprint arXiv:1811.02790*, 2018.
- [199] David Marr. Vision: A computational investigation into the human representation and processing of visual information. 1982.
- [200] Stefan Mathe and Cristian Sminchisescu. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1408–1424, 2014.
- [201] Jonathan Samir Matthis, Jacob L Yates, and Mary M Hayhoe. Gaze and the control of foot placement when walking in natural terrain. *Current Biology*, 28(8):1224–1233, 2018.
- [202] John HR Maunsell. Neuronal mechanisms of visual attention. *Annual Review of Vision Science*, 1:373–391, 2015.
- [203] John HR Maunsell and Stefan Treue. Feature-based attention in visual cortex. *Trends in neurosciences*, 29(6):317–322, 2006.
- [204] Douglas McLelland and Rufin VanRullen. Theta-gamma coding meets communication-through-coherence: neuronal oscillatory multiplexing theories reconciled. *PLoS computational biology*, 12(10), 2016.
- [205] Josh Merel, Diego Aldarondo, Jesse Marshall, Yuval Tassa, Greg Wayne, and Bence Ölveczky. Deep neuroethology of a virtual rodent. *arXiv preprint arXiv:1911.09451*, 2019.

- [206] Georgios Michalareas, Julien Vezoli, Stan Van Pelt, Jan-Mathijs Schoffelen, Henry Kennedy, and Pascal Fries. Alpha-beta and gamma rhythms subserve feedback and feedforward influences among human visual cortical areas. *Neuron*, 89(2):384–397, 2016.
- [207] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.
- [208] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212, 2014.
- [209] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [210] Ida Momennejad, Evan M Russek, Jin H Cheong, Matthew M Botvinick, ND Daw, and Samuel J Gershman. The successor representation in human reinforcement learning. *Nature Human Behaviour*, 1(9):680, 2017.
- [211] Marcelo A Montemurro, Malte J Rasch, Yusuke Murayama, Nikos K Logothetis, and Stefano Panzeri. Phase-of-firing coding of natural visual stimuli in primary visual cortex. *Current biology*, 18(5):375–380, 2008.

- [212] Steven Moore and John C Stamper. Exploring expertise through visualizing agent policies and human strategies in open-ended games. In *EDM (Workshops)*, pages 30–37, 2019.
- [213] Jeffrey Moran and Robert Desimone. Selective attention gates visual processing in the extrastriate cortex. *Science*, 229(4715):782–784, 1985.
- [214] Alexander Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo Jimenez Rezende. Towards interpretable reinforcement learning using attention augmented agents. In *Advances in Neural Information Processing Systems*, pages 12329–12338, 2019.
- [215] Sajad Mousavi, Michael Schukat, Enda Howley, Ali Borji, and Nasser Mozayani. Learning to predict where to look in interactive environments using deep recurrent q-learning. *arXiv preprint arXiv:1612.05753*, 2016.
- [216] Satya M Muddamsetty, Mohammad NS Jahromi, Andreea E Ciontos, Laura M Fenoy, and Thomas B Moeslund. Introducing and assessing the explainable ai (xai) method: Sidu. *arXiv preprint arXiv:2101.10710*, 2021.
- [217] Zoltan Nadasdy. Information encoding and reconstruction from the phase of action potentials. *Frontiers in systems neuroscience*, 3:6, 2009.
- [218] Anis Najar, Olivier Sigaud, and Mohamed Chetouani. Interactively shaping robot behaviour with unlabeled human instructions. *Auton. Agents Multi Agent Syst.*, 34(2):35, 2020.

- [219] Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann Publishers Inc., 2000.
- [220] Tam V Nguyen, Qi Zhao, and Shuicheng Yan. Attentive systems: A survey. *International Journal of Computer Vision*, 126(1):86–110, 2018.
- [221] Scott Niekum, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G Barto. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, 34(2):131–157, 2015.
- [222] Dmitry Nikulin, Anastasia Ianina, Vladimir Aliev, and Sergey Nikolenko. Free-lunch saliency via attention in atari agents. *arXiv preprint arXiv:1908.02511*, 2019.
- [223] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pages 2863–2871, 2015.
- [224] John O’Keefe and Michael L Recce. Phase relationship between hippocampal place units and the eeg theta rhythm. *Hippocampus*, 3(3):317–330, 1993.

- [225] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.
- [226] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.
- [227] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends<sup>®</sup> in Robotics*, 7(1-2):1–179, 2018.
- [228] Andrea Palazzi, Davide Abati, Simone Calderara, Francesco Solera, and Rita Cucchiara. Predicting the driver’s focus of attention: the dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [229] Oskar Palinko, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5048–5054. IEEE, 2016.
- [230] Stefano Panzeri, Jakob H Macke, Joachim Gross, and Christoph Kayser. Neural population coding: combining insights from microscopic and mass signals. *Trends in cognitive sciences*, 19(3):162–172, 2015.

- [231] Dim P Papadopoulos, Alasdair DF Clarke, Frank Keller, and Vittorio Ferrari. Training object class detectors from eye tracking data. In *European conference on computer vision*, pages 361–376. Springer, 2014.
- [232] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediya Daskalova, Jeff Huang, and James Hays. Webgazer: scalable webcam eye tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3839–3845, 2016.
- [233] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep pictorial gaze estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 721–738, 2018.
- [234] Harold Pashler. Dual-task interference in simple tasks: data and theory. *Psychological bulletin*, 116(2):220, 1994.
- [235] Svetlin Penkov, Alejandro Bordallo, and Subramanian Ramamoorthy. Physical symbol grounding and instance learning through demonstration and eye tracking. *International Conference on Automation and Robotics (ICRA)*, 2017.
- [236] Quentin Perrenoud, Cyriel MA Pennartz, and Luc J Gentet. Membrane potential dynamics of spontaneous and visually evoked gamma activity in v1 of awake mice. *PLoS biology*, 14(2):e1002383, 2016.

- [237] Jeffrey S Perry and Wilson S Geisler. Gaze-contingent real-time simulation of arbitrary visual fields. In *Electronic Imaging 2002*, pages 57–69. International Society for Optics and Photonics, 2002.
- [238] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, pages 305–313, 1989.
- [239] Michael I Posner. Orienting of attention. *Quarterly journal of experimental psychology*, 32(1):3–25, 1980.
- [240] James C Prechtl, Theodore H Bullock, and David Kleinfeld. Direct evidence for local oscillatory current sources and intracortical phase gradients in turtle visual cortex. *Proceedings of the National Academy of Sciences*, 97(2):877–882, 2000.
- [241] Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 77–95. Springer, 2020.
- [242] Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. In *International Conference on Learning Representations*, 2019.

- [243] LAI Qiuxia, Salman Khan, Yongwei Nie, Sun Hanqiu, Jianbing Shen, and Ling Shao. Understanding more about human and machine attention in deep neural networks. *IEEE Transactions on Multimedia*, 2020.
- [244] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2586–2591. Morgan Kaufmann Publishers Inc., 2007.
- [245] Harish Chaandar Ravichandar, Avnish Kumar, and Ashwin Dani. Gaze and motion information fusion for human intention inference. *International Journal of Intelligent Robotics and Applications*, 2(2):136–148, 2018.
- [246] Supratim Ray and John HR Maunsell. Do gamma oscillations play a role in cerebral cortex? *Trends in cognitive sciences*, 19(2):78–85, 2015.
- [247] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.
- [248] Charles Rich, Brett Ponsler, Aaron Holroyd, and Candace L Sidner. Recognizing engagement in human-robot interaction. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 375–382. IEEE, 2010.

- [249] Viktor Richter, Birte Carlmeyer, Florian Lier, Sebastian Meyer zu Borgsen, David Schlangen, Franz Kummert, Sven Wachsmuth, and Britta Wrede. Are you talking to me? improving the robustness of dialogue systems in a multi party hri scenario by incorporating gaze direction and lip movement of attendees. In *Proceedings of the Fourth International Conference on Human Agent Interaction*, pages 43–50, 2016.
- [250] DA Robinson. The mechanics of human saccadic eye movement. *The Journal of physiology*, 174(2):245–264, 1964.
- [251] Pieter R Roelfsema, Andreas K Engel, Peter König, and Wolf Singer. Oscillations and synchrony in the visual cortex: evidence for their functional relevance. In *Oscillatory event-related brain dynamics*, pages 99–114. Springer, 1994.
- [252] Pieter R Roelfsema, Victor AF Lamme, and Henk Spekreijse. The implementation of visual routines. *Vision research*, 40(10-12):1385–1411, 2000.
- [253] Khashayar Rohanimanesh and Sridhar Mahadevan. Coarticulation: An approach for generating concurrent plans in markov decision processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 720–727. ACM, 2005.
- [254] Adina L Roskies. The binding problem. *Neuron*, 24(1):7–9, 1999.

- [255] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.
- [256] Stéphane Ross, Geoffrey J Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.
- [257] Constantin A Rothkopf and Dana H Ballard. Image statistics at the point of gaze during human navigation. *Visual neuroscience*, 26(01):81–92, 2009.
- [258] Constantin A Rothkopf and Dana H Ballard. Modular inverse reinforcement learning for visuomotor behavior. *Biological cybernetics*, 107(4):477–490, 2013.
- [259] Constantin A Rothkopf, Dana H Ballard, and Mary M Hayhoe. Task and context determine where you look. *Journal of vision*, 7(14):16–16, 2007.
- [260] Joshua S Rubinstein, David E Meyer, and Jeffrey E Evans. Executive control of cognitive processes in task switching. *Journal of experimental psychology: human perception and performance*, 27(4):763, 2001.
- [261] Kerstin Ruhland, Christopher E Peters, Sean Andrist, Jeremy B Badler, Norman I Badler, Michael Gleicher, Bilge Mutlu, and Rachel McDonnell.

- A review of eye gaze in virtual agents, social robotics and hci: Behaviour generation, user interaction and perception. In *Computer graphics forum*, volume 34, pages 299–326. Wiley Online Library, 2015.
- [262] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [263] Stuart J Russell and Andrew Zimdars. Q-decomposition for reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 656–663, 2003.
- [264] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.
- [265] Kazuyuki Samejima, Kenji Doya, and Mitsuo Kawato. Inter-module credit assignment in modular reinforcement learning. *Neural Networks*, 16(7):985–994, 2003.
- [266] Akanksha Saran, Srinjoy Majumdar, Elaine Schaertl Short, Andrea Thomaz, and Scott Niekum. Human gaze following for human-robot interaction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8615–8621. IEEE, 2018.
- [267] Akanksha Saran, Elaine Schaertl Short, Andrea Thomaz, and Scott

- Niekum. Understanding teacher gaze patterns for robot learning. *Conference on Robot Learning (CoRL)*, 2019.
- [268] Akanksha Saran, Ruohan Zhang, Elaine Schaertl Short, and Scott Niekum. Efficiently guiding imitation learning agents with human gaze. *arXiv preprint arXiv:2002.12500*, 2020.
- [269] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [270] Nicolas Schweighofer, Mathieu Bertin, Kazuhiro Shishida, Yasumasa Okamoto, Saori C Tanaka, Shigeto Yamawaki, and Kenji Doya. Low-serotonin levels increase delayed reward discounting in humans. *the Journal of Neuroscience*, 28(17):4528–4532, 2008.
- [271] Terrence J Sejnowski and Ole Paulsen. Network oscillations: emerging computational principles. *Journal of Neuroscience*, 26(6):1673–1676, 2006.
- [272] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

- [273] Michael N Shadlen and William T Newsome. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience*, 18(10):3870–3896, 1998.
- [274] Ali Shafti, Pavel Orlov, and A Aldo Faisal. Gaze-based, context-aware robotic system for assisted reaching and grasping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 863–869. IEEE, 2019.
- [275] Yang Shen, Bingbing Ni, Zefan Li, and Ning Zhuang. Egocentric activity prediction via event modulated attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 197–212, 2018.
- [276] Wenjie Shi, Zhuoyuan Wang, Shiji Song, and Gao Huang. Self-supervised discovering of causal features: Towards interpretable reinforcement learning. *arXiv preprint arXiv:2003.07069*, 2020.
- [277] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.
- [278] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

- [279] David Silver, J Andrew Bagnell, and Anthony Stentz. Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, 29(12):1565–1592, 2010.
- [280] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [281] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [282] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.
- [283] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

- [284] Wolf Singer. Binding by synchrony. *Scholarpedia*, 2(12):1657, 2007.
- [285] William E Skaggs, Bruce L McNaughton, Matthew A Wilson, and Carol A Barnes. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus*, 6(2):149–172, 1996.
- [286] Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 1938.
- [287] Alec Solway, Carlos Diuk, Natalia Córdova, Debbie Yee, Andrew G Barto, Yael Niv, and Matthew M Botvinick. Optimal behavioral hierarchy. *PLoS computational biology*, 10(8):e1003779, 2014.
- [288] Nathan Sprague and Dana Ballard. Multiple-goal reinforcement learning with modular sarsa (o). In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 1445–1447. Morgan Kaufmann Publishers Inc., 2003.
- [289] Nathan Sprague, Dana Ballard, and Al Robinson. Modeling embodied visual behaviors. *ACM Transactions on Applied Perception (TAP)*, 4(2):11, 2007.
- [290] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

- [291] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [292] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- [293] Giles W Story, Ivo Vlaev, Ben Seymour, Ara Darzi, and Raymond J Dolan. Does temporal discounting explain unhealthy behavior? a systematic review and reinforcement learning perspective. *Frontiers in behavioral neuroscience*, 8, 2014.
- [294] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.
- [295] Felipe Petroski Such, Vashisht Madhavan, Rosanne Liu, Rui Wang, Pablo Samuel Castro, Yulun Li, Jiale Zhi, Ludwig Schubert, Marc G Bellemare, Jeff Clune, et al. An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3260–3267. AAAI Press, 2019.

- [296] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9954–9963, 2019.
- [297] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.
- [298] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*. MIT Press, 1998.
- [299] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [300] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211, 1999.
- [301] Saori C Tanaka, Kenji Doya, Go Okada, Kazutaka Ueda, Yasumasa Okamoto, and Shigeto Yamawaki. Prediction of immediate and future rewards differentially recruits cortico-basal ganglia loops. *Nature neuroscience*, 7(8):887, 2004.
- [302] Benjamin W Tatler, Mary M Hayhoe, Michael F Land, and Dana H Ballard. Eye guidance in natural vision: Reinterpreting salience. *Journal of vision*, 11(5):5–5, 2011.

- [303] Hamed R Tavakoli, Rakshith Shetty, Ali Borji, and Jorma Laaksonen. Paying attention to descriptions generated by image captioning models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2487–2496, 2017.
- [304] Hamed Rezazadegan Tavakoli, Esa Rahtu, Juho Kannala, and Ali Borji. Digging deeper into egocentric gaze prediction. In *2019 IEEE Winter Conference on Applications of Computer Vision*, pages 273–282. IEEE, 2019.
- [305] Andrea Thomaz, Guy Hoffman, Maya Cakmak, et al. Computational human-robot interaction. *Foundations and Trends® in Robotics*, 4(2-3):105–223, 2016.
- [306] Matthew H. Tong, Mary M. Hayhoe, Oran Zohar, Ruohan Zhang, Dana H. Ballard, and Shun Zhang. Multitask Human Navigation in VR with Motion Tracking, January 2017.
- [307] Matthew H Tong, Oran Zohar, and Mary M Hayhoe. Control of gaze while walking: task structure, reward, and uncertainty. *Journal of Vision*, 2017.
- [308] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4950–4957. AAAI Press, 2018.

- [309] Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6325–6331. AAAI Press, 2019.
- [310] Pedro A Tsividis, Thomas Pouncy, Jacqueline L Xu, Joshua B Tenenbaum, and Samuel J Gershman. Human learning in atari. 2017.
- [311] Timo Van Kerkoerle, Matthew W Self, Bruno Dagnino, Marie-Alice Gariel-Mathis, Jasper Poort, Chris Van Der Togt, and Pieter R Roelfsema. Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex. *Proceedings of the National Academy of Sciences*, 111(40):14332–14341, 2014.
- [312] Timo Van Kerkoerle, Matthew W Self, and Pieter R Roelfsema. Layer-specificity in the effects of attention and working memory on activity in primary visual cortex. *Nature communications*, 8:13804, 2017.
- [313] Stan van Pelt, Lieke Heil, Johan Kwisthout, Sasha Ondobaka, Iris van Rooij, and Harold Bekkering. Beta-and gamma-band activity reflect predictive coding in the processing of causal events. *Social cognitive and affective neuroscience*, 11(6):973–980, 2016.
- [314] Harm Van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5392–5402, 2017.

- [315] Rufin VanRullen and Simon J Thorpe. Surfing a spike wave down the ventral stream. *Vision research*, 42(23):2593–2615, 2002.
- [316] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3540–3549. JMLR. org, 2017.
- [317] Martin Vinck and Conrado Bosman. More gamma, more predictions: Gamma- synchronization supports the integration of classical receptive field inputs with predictions from the surround. *Frontiers in Systems Neuroscience*, 2016.
- [318] Martin Vinck and Conrado A Bosman. More gamma more predictions: gamma-synchronization as a key mechanism for efficient integration of classical receptive field inputs with surround predictions. *Frontiers in systems neuroscience*, 10, 2016.
- [319] Martin Vinck, Bruss Lima, Thilo Womelsdorf, Robert Oostenveld, Wolf Singer, Sergio Neuenschwander, and Pascal Fries. Gamma-phase shifting in awake monkey visual cortex. *Journal of Neuroscience*, 30(4):1250–1257, 2010.
- [320] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo

- Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- [321] Junpeng Wang, Liang Gou, Han-Wei Shen, and Hao Yang. Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):288–298, 2018.
- [322] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4894–4903, 2018.
- [323] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1995–2003, 2016.
- [324] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [325] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

- [326] Nicholas R Waytowich, Vinicius G Goecks, and Vernon J Lawhern. Cycle-of-learning for autonomous systems from human interaction. *arXiv preprint arXiv:1808.09572*, 2018.
- [327] Laurens Weitkamp, Elise van der Pol, and Zeynep Akata. Visual rationalizations in deep reinforcement learning for atari games. In *Benelux Conference on Artificial Intelligence*, pages 151–165. Springer, 2018.
- [328] Erik Wijmans, Julian Straub, Dhruv Batra, Irfan Essa, Judy Hoffman, and Ari Morcos. Analyzing visual representations in embodied navigation tasks. *arXiv preprint arXiv:2003.05993*, 2020.
- [329] Daniel M Wolpert and Michael S Landy. Motor control is decision-making. *Current opinion in neurobiology*, 22(6):996–1003, 2012.
- [330] Thilo Womelsdorf, Pascal Fries, Partha P Mitra, and Robert Desimone. Gamma-band synchronization in visual cortex predicts speed of change detection. *Nature*, 439(7077):733, 2006.
- [331] Thilo Womelsdorf, Jan-Mathijs Schoffelen, Robert Oostenveld, Wolf Singer, Robert Desimone, Andreas K Engel, and Pascal Fries. Modulation of neuronal interactions through neuronal synchronization. *science*, 316(5831):1609–1612, 2007.
- [332] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape regis-

- tration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3756–3764, 2015.
- [333] Mark Woodward, Chelsea Finn, and Karol Hausman. Learning to interactively learn and assist. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2535–2543, 2020.
- [334] Ye Xia, Jinkyu Kim, John Canny, Karl Zipser, Teresa Canas-Bajo, and David Whitney. Periphery-fovea multi-resolution driving model guided by human attention. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1767–1775, 2020.
- [335] Ye Xia, Jinkyu Kim, John Canny, Karl Zipser, and David Whitney. Periphery-fovea multi-resolution driving model guided by human attention. *arXiv preprint arXiv:1903.09950*, 2019.
- [336] Ye Xia, Danqing Zhang, Jinkyu Kim, Ken Nakayama, Karl Zipser, and David Whitney. Predicting driver attention in critical situations. In *Asian conference on computer vision*, pages 658–674. Springer, 2018.
- [337] Juan Xu, Ming Jiang, Shuo Wang, Mohan S Kankanhalli, and Qi Zhao. Predicting human gaze beyond pixels. *Journal of vision*, 14(1):28–28, 2014.
- [338] Yanyu Xu, Yanbing Dong, Junru Wu, Zhengzhong Sun, Zhiru Shi, Jingyi Yu, and Shenghua Gao. Gaze prediction in dynamic 360 immersive

- videos. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5333–5342, 2018.
- [339] Daniel L Yamins, Ha Hong, Charles Cadieu, and James J DiCarlo. Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. In *Advances in Neural Information Processing Systems*, pages 3093–3101, 2013.
- [340] Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.
- [341] Zhao Yang, Song Bai, Li Zhang, and Philip HS Torr. Learn to interpret atari agents. *arXiv preprint arXiv:1812.11276*, 2018.
- [342] Chunxiu Yu, Guy Horev, Naama Rubin, Dori Derdikman, Sebastian Haidarliu, and Ehud Ahissar. Coding of object location in the vibrissal thalamocortical system. *Cerebral Cortex*, 25(3):563–577, 2013.
- [343] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

- [344] Mingxin Yu, Yingzi Lin, David Schmidt, Xiangzhou Wang, and Yu Wang. Human-robot interaction based on gaze gestures for the drone teleoperation. *Journal of Eye Movement Research*, 7(4):1–14, 2014.
- [345] Liu Yuezhong, Ruohan Zhang, and Dana H Ballard. An initial attempt of combining visual selective attention with deep reinforcement learning. *arXiv preprint arXiv:1811.04407*, 2018.
- [346] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
- [347] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014.
- [348] Gregory Zelinsky, Zhibo Yang, Lihan Huang, Yupei Chen, Seoyoung Ahn, Zijun Wei, Hossein Adeli, Dimitris Samaras, and Minh Hoai. Benchmarking gaze prediction for categorical visual search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [349] Alexandre Zénon, Yann Duclos, Romain Carron, Tatiana Witjas, Christelle Baunez, Jean Régis, Jean-Philippe Azulay, Peter Brown, and Alexandre Eusebio. The human subthalamic nucleus encodes the subjective value of reward and the cost of effort during decision-making. *Brain*, 139(6):1830–1843, 2016.

- [350] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.
- [351] Luxin Zhang, Ruohan Zhang, Zhuode Liu, Mary M Hayhoe, and Dana H Ballard. Learning attention model from human for visuomotor tasks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [352] R Zhang, A Saran, B Liu, Y Zhu, S Guo, S Niekum, D Ballard, and M Hayhoe. Human gaze assisted artificial intelligence: A review. In *International Joint Conference on Artificial Intelligence*, 2020.
- [353] Ruohan Zhang. Action selection in modular reinforcement learning. *Master Thesis*, 2014.
- [354] Ruohan Zhang. Attention guided imitation learning and reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9906–9907, 2019.
- [355] Ruohan Zhang and Dana H Ballard. Parallel neural multiprocessing with gamma frequency latencies. *Neural Computation*, 32(9):1635–1663, 2020.
- [356] Ruohan Zhang, Bo Liu, Yifeng Zhu, Sihang Guo, Mary Hayhoe, Dana Ballard, and Peter Stone. Human versus machine attention in deep reinforcement learning tasks. *arXiv preprint arXiv:2010.15942*, 2020.

- [357] Ruohan Zhang, Zhuode Liu, Mary M Hayhoe, and Dana H Ballard. Attention guided deep imitation learning. In *Cognitive Computational Neuroscience (CCN)*, 2017.
- [358] Ruohan Zhang, Zhuode Liu, Luxin Zhang, Karl S Muller Mary M Hayhoe, and Dana H Ballard. Visual attention guided deep imitation learning. *Advances in neural information processing systems workshop*, 2017.
- [359] Ruohan Zhang, Zhuode Liu, Luxin Zhang, Jake A Whritner, Karl S Muller, Mary M Hayhoe, and Dana H Ballard. Agil: Learning attention from human for visuomotor tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 663–679, 2018.
- [360] Ruohan Zhang, Zhao Song, and Dana H Ballard. Global policy construction in modular reinforcement learning. In *AAAI*, pages 4226–4227, 2015.
- [361] Ruohan Zhang, Faraz Torabi, Lin Guan, Dana H Ballard, and Peter Stone. Leveraging human guidance for deep reinforcement learning tasks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6339–6346. AAAI Press, 2019.
- [362] Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl Muller, Jake Whritner, Luxin Zhang, Mary Hayhoe, and Dana Ballard. Atari-head: Atari human eye-tracking and demonstration dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6811–6820, 2020.

- [363] Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl S Muller, Jake A Whritner, Luxin Zhang, Mary M Hayhoe, and Dana H Ballard. Atari-head: Atari human eye-tracking and demonstration dataset. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2020.
- [364] Ruohan Zhang, Jake Whritner, Zhuode Liu, Luxin Zhang, Karl Muller, Mary Hayhoe, and Dana Ballard. Modelling complex perception-action choices. *Journal of Vision*, 18(10):533–533, 2018.
- [365] Ruohan Zhang, Yue Yu, Mahmoud El Chamie, Behçet Açıkmeşe, and Dana H Ballard. Decision-making policies for heterogeneous autonomous multi-agent systems with safety constraints. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 546–552. AAAI Press, 2016.
- [366] Ruohan Zhang, Shun Zhang, Matthew Tong, Mary Hayhoe, and Dana Ballard. Modeling sensorimotor behavior through modular inverse reinforcement learning with discount factors. *Journal of Vision*, 17(10):1267–1267, 2017.
- [367] Ruohan Zhang, Shun Zhang, Matthew H Tong, Yuchen Cui, Constantin A Rothkopf, Dana H Ballard, and Mary M Hayhoe. Modeling sensory-motor decisions in natural behavior. *PLoS computational biology*, 14(10):e1006518, 2018.

- [368] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015.
- [369] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpi-gaze real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017.
- [370] Ziheng Zhang, Yanyu Xu, Jingyi Yu, and Shenghua Gao. Saliency detection in 360 videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 488–503, 2018.
- [371] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.
- [372] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd national conference on Artificial intelligence-Volume 3*, pages 1433–1438. AAAI Press, 2008.
- [373] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

- [374] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.
- [375] Zheming Zuo, Longzhi Yang, Yonghong Peng, Fei Chao, and Yanpeng Qu. Gaze-informed egocentric action recognition for memory aid systems. *IEEE Access*, 6:12894–12904, 2018.

## Vita

Ruohan Zhang was born in Chengdu, China, July 1988. He graduated from Chengdu No.7 High School. In 2008 he came to the United States to study at Rhodes College, Memphis, TN. In May 2012 he received a B.A. degree in Psychology, with a Minor in Computer Science, and a Minor in Economics. In August 2014, he received an M.S. degree in Computer Science at the University of Texas at Austin. He continued his Ph.D. study at the University of Texas at Austin. His research interests are computational neuroscience, reinforcement learning, and robotics.

Permanent address: 3400 Speedway 309  
Austin, Texas 78705

This dissertation was typeset with L<sup>A</sup>T<sub>E</sub>X<sup>†</sup> by the author.

---

<sup>†</sup>L<sup>A</sup>T<sub>E</sub>X is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T<sub>E</sub>X Program.