Social Inductive Biases for Reinforcement Learning

by

Dhaval D.K. Adjodah

B.S., Physics (2011), Massachusetts Institute of Technology
M.S., Technology and Policy (2011), Massachusetts Institute of Technology

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

Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2019

Signature redacted

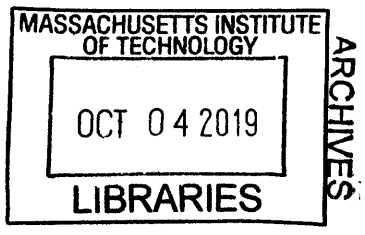
Author .....
Program in Media Arts and Sciences,
School of Architecture and Planning,
July 2, 2019

Signature redacted

Certified by .....
Alex "Sandy" P. Pentland
Toshiba Professor of Media Arts and Sciences
Thesis Supervisor

Signature redacted

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



Social Inductive Biases for Reinforcement Learning

by

Dhaval D.K. Adjodah

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on July 2, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

Abstract

How can we build machines that collaborate and learn more seamlessly with humans, and with each other? How do we create fairer societies? How do we minimize the impact of information manipulation campaigns, and fight back? How do we build machine learning algorithms that are more sample efficient when learning from each other's sparse data, and under time constraints? At the root of these questions is a simple one: how do agents, human or machines, learn from each other, and can we improve it and apply it to new domains?

The cognitive and social sciences have provided innumerable insights into how people learn from data using both passive observation and experimental intervention. Similarly, the statistics and machine learning communities have formalized learning as a rigorous and testable computational process.

There is a growing movement to apply insights from the cognitive and social sciences to improving machine learning, as well as opportunities to use machine learning as a sandbox to test, simulate and expand ideas from the cognitive and social sciences. A less researched and fertile part of this intersection is the modeling of social learning: past work has been more focused on how agents can learn from the 'environment', and there is less work that borrows from both communities to look into how agents learn from each other.

This thesis presents novel contributions into the nature and usefulness of social learning as an inductive bias for reinforced learning. I start by presenting the results from two large-scale online human experiments: first, I observe Dunbar cognitive limits that shape and limit social learning in two different social trading platforms, with the additional contribution that synthetic financial bots that transcend human limitations can obtain higher profits even when using naive trading strategies. Second, I devise a novel online experiment to observe how people, at the individual level, update their belief of future financial asset prices (e.g. S&P 500 and Oil prices) from social information. I model such social learning using Bayesian models of cognition, and observe that people make strong distributional assumptions on the social data

they observe (e.g. assuming that the likelihood data is unimodal). I were fortunate to collect one round of predictions during the Brexit market instability, and find that social learning leads to higher performance than when learning from the underlying price history (the environment) during such volatile times. Having observed the cognitive limits and biases people exhibit when learning from other agents, I present an motivational example of the strength of inductive biases in reinforcement learning: I implement a learning model with a relational inductive bias that pre-processes the environment state into a set of relationships between entities in the world. I observe strong improvements in performance and sample efficiency, and even observe the learned relationships to be strongly interpretable. Finally, given that most modern deep reinforcement learning algorithms are distributed (in that they have separate learning agents), I investigate the hypothesis that viewing deep reinforcement learning as a social learning distributed search problem could lead to strong improvements. I do so by creating a fully decentralized, sparsely-communicating and scalable learning algorithm, and observe strong learning improvements with lower communication bandwidth usage (between learning agents) when using communication topologies that naturally evolved due to social learning in humans. Additionally, I provide a theoretical upper bound (that agrees with our empirical results) regarding which communication topologies lead to the largest learning performance improvement.

Given a future increasingly filled with decentralized autonomous machine learning systems that interact with humans, there is an increasing need to understand social learning to build resilient, scalable and effective learning systems, and this thesis provides insights into how to build such systems.

Thesis Supervisor: Alex "Sandy" P. Pentland
Title: Toshiba Professor of Media Arts and Sciences

Social Inductive Biases for
Reinforcement Learning
by
Dhaval D.K. Adjodah

Signature redacted

Thesis Advisor .....
Alex "Sandy" P. Pentland
Toshiba Professor of Media Arts and Sciences
MIT Program in Media Arts and Sciences

Thesis reader .....
Neil Lawrence
Professor of Machine Learning
Chair in Neuroscience and Computer Science
Department of Computer Science
The University of Sheffield

Signature redacted

Thesis reader .....
Tim Klinger
Research Staff Member
IBM Research AI
IBM Thomas J. Watson Research Center

Thesis reader .....
Esteban Moro
Full Professor of the University
Department of Mathematics
Universidad Carlos III de Madrid

Social Inductive Biases for
Reinforcement Learning
by
Dhaval D.K. Adjodah

Thesis Advisor .....
Alex "Sandy" P. Pentland
Toshiba Professor of Media Arts and Sciences
MIT Program in Media Arts and Sciences

Signature redacted

Thesis reader .....
Neil Lawrence
Professor of Machine Learning
Chair in Neuroscience and Computer Science
Department of Computer Science
The University of Sheffield

Thesis reader .....
Tim Klinger
Research Staff Member
IBM Research AI
IBM Thomas J. Watson Research Center

Thesis reader .....
Esteban Moro
Full Professor of the University
Department of Mathematics
Universidad Carlos III de Madrid

Social Inductive Biases for
Reinforcement Learning
by
Dhaval D.K. Adjodah

Signature redacted

Thesis Advisor .....

Alex "Sandy" P. Pentland
Toshiba Professor of Media Arts and Sciences
MIT Program in Media Arts and Sciences

Thesis reader .....

Neil Lawrence
Professor of Machine Learning
Chair in Neuroscience and Computer Science
Department of Computer Science
The University of Sheffield

Thesis reader .....

Tim Klinger
Research Staff Member
IBM Research AI
IBM Thomas J. Watson Research Center

Signature redacted

Thesis reader .....

Esteban Moro
Full Professor of the University
Department of Mathematics
Universidad Carlos III de Madrid

Acknowledgments

I often ponder how I ended up being able to start a career focused on thinking about, and unravelling the mysteries of the universe. I think I owe it to a lot of luck, some laziness, some existential dread, and a lot to the people that have been there to support me throughout this journey.

Some of these people are unfortunately no longer here for me to excitedly share my journey with them. Maman who, through herculean efforts, carried our family through unbelievable odds: I wish you were still in this world for me to tell you what I am working on. But as you told me, if we have been together at some point in the history of the universe, we are always together. Thank you. Sam Bowring, who, during my first semester of my freshman year at MIT, gave me a literal and figurative microphone to speak my mind, and call on other people's bullshit: I wish I could thank you for believing in me, and choosing to accompany me in my various mischief to make sure I was okay, instead of getting me in trouble for it. Mr. and Mrs. Tracol, thank you for showing me what it is like to live a life with little social compromise, and showing me that even people from faraway lands and different ages can become great friends. To them, who are no longer with me, I keep you in my heart.

I still remember fondly the many times my dad taught me to break the rules - going fishing at the river, playing with electronics to make them do things they were not supposed to. Thank you for showing me that sometimes rules were meant to be broken. To my brother, thank you for sharing your passion for asking questions, tinkering with broken down VCRs, building a sustenable pond ecosystem in our backyard, poking under the layers of the lies they told us at school. You are a riot.

Alia, the love of my life, I would also like to thank you for your support, love, patience and your ability to call me on my bullshit. I would also like to thank my family-in-law for their love, and making me one of their own. I do think Kaila has a pottery problem though.

Instrumental to my PhD, I would like to thank my PhD committee. Sandy Pent-

land, my main advisor, thank you for taking in a wide-eyed intellectual refuge from across MIT and giving me a chance, and giving me the academic freedom to ask new questions. I cannot thank you enough for your support. Esteban Moro, who I met during a casual lunch where he wanted to deeply understand why humans behave the way they do, thank you for enlightening me to the effect of cognitive constraints on reasoning. Tim Klinger who I met during my internship at IBM, thank you for your many hours of brainstorming, discussions about relational model structures, and for your strong collaboration even after my internship ended. I knew from our first conversation during that lunch in the Yorktown cafetaria that I wanted to work with you because you just questionned 400 years of rational thought. I learned immensely from you. Neil Lawrence, who I met on the dance floor on a party-on-a-ship, thank you for your many hours brainstorming about Theory of Mind, and how to build better learning algorithms for human-machine collaboration.

Of course, I would also like to thank my many friends and colleagues in the Human Dynamics group at the MIT Media Lab; as well as Dong Ki Kim and Shayegan Omidshafiei of the MIT Aero-Astro ACL group; Yoshua Bengio and Anirudh Goyal at MILA in Montreal, Murray Campbell at IBM. I'd like to also thank the many amazing people at the MIT Media Lab who have helped me along the way, especially Linda Peterson, Joi Ito, Keira Horowitz, Julie Hall, Nicole Freedman, Natalie Saltiel, James Frost.

Special thanks to the many amazing friends at MIT that I met along the way, especially Ray Reich, Bayo Olatunji, Adam Schortman, Josh Joseph, John Hess, Camille McAvoy, Tal Achituv, Shivraj Sohur, Debb Hodges-Pabon, Benjy Leinwand, Brandon Sorbom, Jessica Artiles.

Finally, I would like to thank my many friends and family in Mauritius for their love and support, especially Harish Adjodah, Dharvish Chundidyal, Satyam Veerateraapillay, Vij Bheenick, Chetan Boodhoo, Benu and Sheckhar Servansingh, Danen Sungeelee, Indiren Gohiden, Ryan Khoodoruth, Rishikesh Doobaree, Dolaree Nuckcheddy.

I can't end without thanking the countless faceless people on stackoverflow, wikipedia,

reddit, quora, youtube, etc. who disseminated knowledge like never before.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The cognitive and social sciences have provided innumerable insights into how people learn from data using both passive observation and experimental intervention. Similarly, the statistics and machine learning communities have formalized learning as a rigorous and testable computational process. There is a growing movement to apply insights from the cognitive and social sciences to improving machine learning [65], as well as opportunities to use machine learning as a sandbox to test, simulate and expand ideas from the cognitive and social sciences [28, 93]. A less researched and fertile part of this intersection is the modeling of social learning: past work has been mostly focused on how agents can learn from the 'environment', and there is little work that borrows from both communities to look into how agents learn from each other.

This thesis presents novel contributions into the nature and usefulness of social learning.

1.1 Problem Statement

How can we build machines that collaborate and learn more seamlessly with humans, and with each other? How do we create fairer societies? How do we minimize the impact of information manipulation campaigns, and fight back? How do we build machine learning algorithms that are more sample efficient when learning from each

other's sparse data, and under time constraints?

At the root of these questions is a simple one: **how do agents, human or machine, learn from each other, and can we improve it and apply it to new domains?**

Although there has been a huge amount of research ([8, 27, 67, 118]) on the question – especially in the wake of massive amount of interactions data from online social network – we still know so little about how learning from social information is undertaken by the most intelligent and social creatures we know of: humans.

This thesis hopes to add insights into this question. First, I report results from two massive online studies where I investigate at unprecedented scale and precision the cognitive limitations and inductive biases humans exhibit when learning from each other, and the trade-offs in doing so. I then apply these insights into machine learning algorithms and models, and show that doing so can vastly increase learning performance.

This is especially important as modern machine learning algorithms are becoming more and more distributed by spawning separate learning instances that are searching the optimization landscape for the best generalizable model parameters.

1.2 Inductive Biases in Learning

The goal of learning is to come up with a model M given some data D through using an algorithm L. The model M is used either by a human to make decisions (e.g. What restaurant should I go to?) or by a machine (Is this a picture of a dog or dolphin?).

How complex should the model be in order to learn from the data? Or how much data is needed to learn the a model with low probability of error? Although these questions have been traditionally formalized using the probably approximately correct (PAC) learning framework[112, 35], here, I discuss a more intuitive (although perhaps less practical) approach.

It can be shown by using algorithmic information theory that the Kolmogorov

complexity of the learned model has to always be at least as large as the sum of the Kolmogorov complexities of the data and the algorithm[53]. The Kolmogorov complexity is a measure of the shortest program that can perfectly generate the required sequence of data, learning algorithm or model.

$$K(M) \geq K(L) + K(D) \tag{1.1}$$

The more structured the data, the smaller will its complexity be, and, consequently, the smaller will be the model that needs to be learned to correct predict the data. For example, if all points in the data can be grouped into two infinitely dense and separated points, then it will be trivial to reproduce and classify the data - any new point can only be in one of two locations. However, if the data was uniformly distributed in the space, then each point might have a unique location and will be much harder to predict.

Thus, if we know that the data has a certain structure, we can 'induce' this structural bias in the learning algorithm or the model by encouraging the correct representation learning[17]. This might cause learning to require less samples to learn (i.e. it will increase the sample efficiency of our machine learning approach). Additionally, using the correct inductive biases can lead to better generalization from training to testing. Although advantageous, this is not strictly needed: given enough time, data, the minimal required loss function and optimizer, it should still be able to learn in theory, but will require more examples. However, in practice, having an algorithm learn faster is preferred and therefore such inductive biases are quite common and sought after.

For example, convolutional neural networks encode the inductive bias of spatial equivariance (i.e. features close to each other are important), recurrent neural networks encode the inductive bias of temporal proximity, graph neural networks encode the inductive bias that the relation between objects are important. In the next section, we discuss the inductive bias of relational logical operators for games where logical operators are a natural language for solving the puzzle at hand. For a recent

review of such inductive biases, please refer to [13].

Inductive biases are especially important because, at its core, machine learning is not aimed at learning concepts in the epistemological sense: a machine learning model is not trying to understand what a 'dog' is beyond trying find patterns in data that correlate with the label 'dog'[1]. For example, in the case of facial recognition, an algorithm does not understand the concept of a human face, but instead learns that over the distribution of pixels, certain patterns of pixels corresponds to faces.

This distinction is important as it is the basis of adversarial machine learning[46], whereby small perturbation in the data can be implemented to cause the model to misclassify its data. For example, an adversarial attack on face detection can be undertaken by injecting a small amount of noise causing the model to not detect a face anymore. In Figure 1-1, we add increasing levels of uniform random noise to images and see that various benchmark models and API's performance degrade quickly. The attack was implemented using Fast Gradient Sign Attack (FGSM)[46].

1.2.1 Inductive Biases in Cognitive Science

Inductive biases in learning and decision making have a rich history in cognitive science, economics and social science where they are called 'biases and heuristics' and are studied mostly as a negative in terms of the irrationalities people exhibit when learning from data. A popular and useful framework to understand human learning and bias is the System 1 and System 2 categorization by Kahneman and Tversky[57, 111]:

- System 1: the unconscious, automatic, fast but error-prone approach to decision. It requires minimal cognitive load and is used for most small decisions.

- System 2: the conscious, deliberate, slow reasoning required for complex, novel decisions. It imposes a large cognitive load.

[1]This is under debate as some machine learning proponents might argue that the correct representation of a dog should be as close to the epistemological concept of a dog one can have[103, 40], but it is unclear how to achieve this.

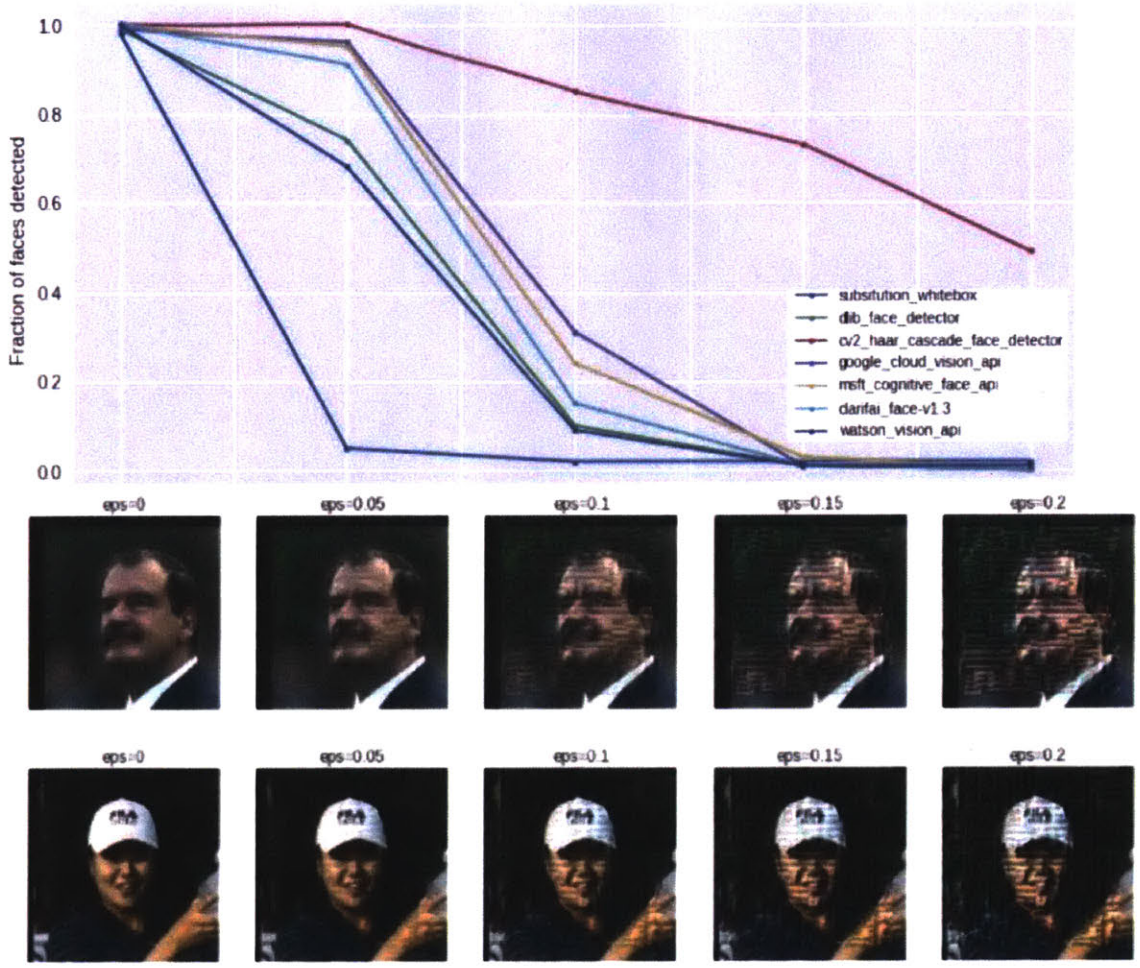


Figure 1-1: Because face detection machine learning algorithm are searching for patterns in the data, adversarial attacks can be implemented easily even on publicly available face-detection API's.

Various heuristic and biases have been documented over the years in how people learn from data, but there has been limited work into how people learn from each other, which is a huge part of our species success. For example, we do not each have to be burned by fire to know that touching fire is bad; we can learn this from other people's prior knowledge. There is even evidence that specific brain regions are responsible for perceiving social information [54] and in reasoning about others' minds[97].

The question, then, is how do people learn from each other's information? This question is more difficult to instrument and investigate as it is harder to know what

kind of information people are looking at when they learn from each other. Are people looking at people's facial expression, their actions, their rewards, or their natural language? There is evidence that individuals do not always give their best guess when answering a question [114], but instead sample from an internal distribution. This question has been studied computationally in Theory of Minds[8], but not much prior has been done from the perspective of inductive biases. We believe that a novel approach to understanding the inductive biases of social learning would be to have access to the distribution of belief of people and model how an individual's belief is updated given this information. This will be the foundation of one of our large studies.

1.3 Thesis outline:

In this section, we describe the structure of this thesis, along with a summary of each chapter.

1.3.1 Cognitive Limits in Social Learning

We start by investigating the cognitive reasons that require the need for inductive biases in human social learning. Using two large datasets of social trading behavior and performance, we observe hard constraints on the amount of social information a trader can learn from. Our work builds on previous studies that have observed the presense of cognitive limits in social interactions in various contexts[73, 74, 45], but they lacked a strong measure of individual performance (which we can measure in our domains using the profit of the traders). In this study, we contribute the observation of a strong connection between cognitive limits and an agent's performance, and identify the strategies humans use to cope with their attention and cognitive limits.

We choose to study financial networks because they are a unique domain where there is a clear performance measure (profit) that depends on a need to learn information (strategies) from other people. It is therefore a very useful domain to test if, even in such a performance-focused environment, cognitive limits on the number of

agents one can attend to exist, and how they affect performance.

1.3.2 Inductive Biases in Social Learning

In chapter 3, given that people have cognitive limits (observed in the previous chapter), we investigate the social inductive biases that emergy as a remedy to the effect of cognitive constraints on social learning.

To shed light into these questions, we collected data through a large online experiment where 2,037 mid-career financial professionals made a total of 9,268 estimates of future financial asset prices (the S&P 500, WTI Oil and gold prices) in a series of seven independent live prediction rounds. By carefully instrumenting the data that each individual is shown after they made their initial pre-exposure prediction of the asset prices, and then recording their revised prediction, we are able to investigate the inductive biases people employ when learning from others.

We do so by modeling the statistical nature of individual and collective estimation in social learning using formalism inspired by Bayesian models of cognition.

1.3.3 Improving Deep Reinforcement Learning at the Individual Level

In chapter 4, we present how augmenting a deep reinforcement learning (DRL) model with relational (between agents and objects) inductive biases can significantly improve performance. Such relational descriptions have been observed to be used by humans in learning social relationships.

In recent years, reinforcement learning techniques have enjoyed considerable success in a variety of challenging domains, but are typically sample inefficient and often fail to generalize well to new environments or tasks. Humans, by contrast, are able to learn robust skills with orders of magnitude less training. One hypothesis for this discrepancy is that humans view the world in terms of objects and relations between them. Such an inductive bias may be useful reducing sample complexity and improving interpretability and generalization. In this chapter, we present a novel relational

architecture which has multiple neural network sub-modules called *relational units* which operate on objects and output values in the unit interval. Our model transforms the input state representation into a relational representation, which is then supplied as input to a Q-learner. Experiments on a simple goal-seeking game show better performance over several baselines: a multi-headed attention model, a standard MLP, a pixel MLP and a symbolic RL model. We also find that the relations learned in the network are interpretable.

In summary, by pre-processing the input state to an off-the-shelf reinforcement algorithm into a relational description using a relational inductive bias, we are able to significantly improve sample efficiency.

1.3.4 Improving Deep Reinforcement Learning at the Collective Level

In chapter 4, we study how, because of cognitive limits, humans *collectively* adapt their communication topologies to improve learning. This is applicable to DRL because modern approaches utilize distributed algorithms that rely on an implicit communication network between the processing units being used in the algorithm, and there is a lot of potential to wire and organize these learning agents less naively.

Given these cognitive limits, humans and animal species have evolved a number of adaptations in order to optimally search a parameter landscape for the best performing set. One of these adaptations is the use of certain sparse topologies[11] of communication as a way to trade-off exploration and exploitation of parameters.

c

Given that network effects are sometimes only significant with large numbers of agents, we choose to build upon one of the DRL algorithms most oriented towards parallelizability and scalability: Evolution Strategies [96, 102, 115, 100]. We introduce Networked Evolution Strategies (NetES), a networked decentralized variant of ES. Using NetES, we explore how the communication topology of a population of processors affects learning performance.

1.4 Summary

The structure of the thesis is as follows: we will first observe the cognitive limits causing humans to require such inductive biases, followed by investigating precisely the effect of these inductive biases on learning from others, and then show how such social inductive biases can be used to improve deep reinforcement learning both at the individual and collective level.

Chapter 2

Cognitive Limits in Social Learning

2.1 Purpose

We start by investigating the cognitive pressures that might lead to the need for inductive biases in the case of human social learning beyond their observed benefits in learning improvement, and we observe the impact of cognitive limits on agent performance, and the strategies humans use to cope with their attention and cognitive limits. We also create bots with simple strategies but very large cognitive capacities and observe that they sometimes have superhuman performances.

2.2 Background

We analyze the dynamics of the mirroring behavior in the eToro[90, 62] and Darwinex trading platforms. In that platform traders can copy ("mirror") other users' trades and we are interested in knowing how do they manage those mirror-ing links in time. Recent literature in different contexts has highlighted the cognitive, time and cost bounds in human tasks and, especially in social interactions [73, 74, 45]. Our cognitive limits affect the maximum number of social connections we can have. Other human activities are also constrained since our attention is limited [51]. The result is that humans have different strategies to cope with their attention and cognitive limits and in particular we end up with a finite number of relationships or tasks that

we can perform per unit time.

The platforms offers an interesting place to test this hypothesis since not only we have data about how traders manage to mirror other users' trades, but also how their performance in trading is coupled with that behavior. In social settings, our previous research on mobile phone data found that people have very different social strategies: while some users keep their connections for a long time without creating/destroying any link (social keepers) other users present a large turnover of their social connections along time (social explorers). But in both cases, users have a fixed number of social connections the can maintain at any instant, what we called *social capacity.*

Our objective is thus to try to answer the following questions with these new datasets:

- Are traders also cognitive limited so that the number of mirroring links they can have is bounded?

- We would like to investigate if different dynamical mirroring strategies have any correlation with their trading performance. Specifically, is keeping the same mirroring connections better than constantly changing them?

- Can we simulate trading bots that allow us to explore cognitive regions beyond the ability of human traders?

2.3 Data

We obtained data from two of the major social trading platforms, Darwinex and eToro, and we compare and constrast trader behavior on the two platforms. In both platforms, traders are able to view the profiles of other traders where various information is shown in real-time such as RoI (return on investment), popularity (e.g. the amount of capital others have invested in this trader, or the number of other traders investing in this user), risk, rankings, and demographic data. A screenshot of the eToro profile summaries is shown in Figure 2-1, and Darwinex in Figure 2-2.

Figure 2-1: Screenshot of the Darwinex "exchange" where a summary of each trader (referred to as a 'darwin') is displayed. Users can invest in a darwin by clicking the 'TRADE' button.

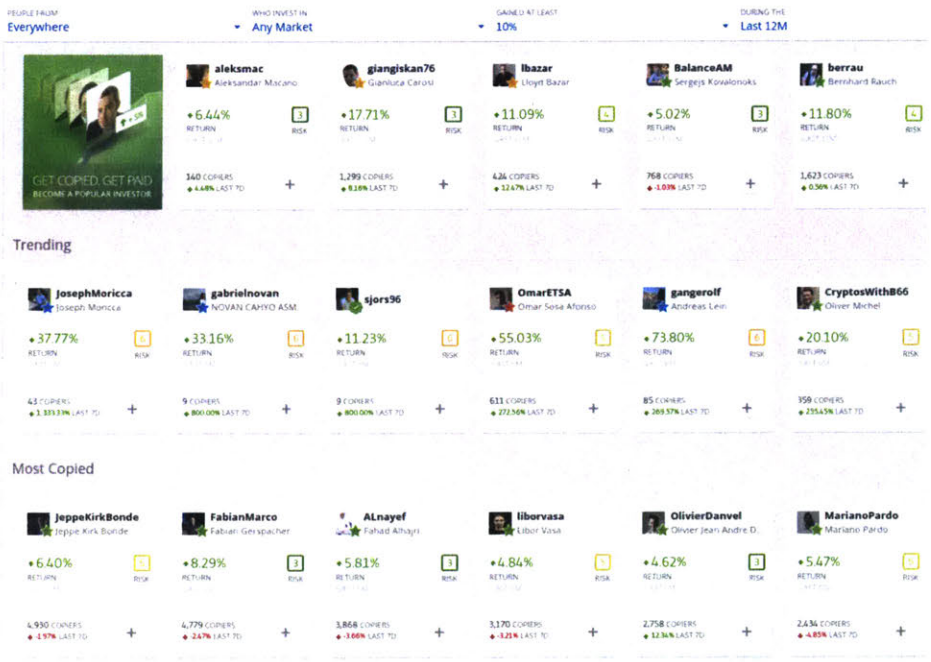


Figure 2-2: Screenshot of the eToro platform where users can choose which other traders to invest in (referred to as 'copying' or 'mirroring' on the platform.)

2.3.1 eToro Dataset

In eToro, users can choose to either buy conventional assets (such as shares of the S&P 500), or they can 'mirror' other traders' trades on conventional assets: a user i can click the 'copy' button on another trader j's profile which would cause a custom percentage of user i's capital to be spent replicating the trades on conventional assets bought and sold by trader j. Later we will build bots that will simulate such 'mirroring' behavior. There are no fees to mirror other traders or to buy conventional assets (eToro's platform revenue comes from spreads).

We have 87,967,223 conventional trades from June 2011 to November 2013 from more than 50,000 traders, where each trade consists of the user id of the trader, the opening and closing dates, the volume, open and close price and profit received from this trade, amongst other information. We calculate the ROI of each trade k as $r_k = \frac{\text{profit}_k}{\text{volume}_k \times \text{open price}_k}$. We compute the ROI of each trader i for a time period T (e.g. one day or one month) as the average ROI over all N conventional trades executed by trader i during period T as $R_{i,T} = \frac{1}{N}\sum_{k=1}^{N} r_k$.

Since we did not have access to the data being displayed on the web-interface of eToro, we used the daily user statistics (such as the popularity and performance of each trader) that were previously reconstructed (refer to [63] for the exact computations as we use their data) to be similar the data displayed on the eToro web interface. We will use this data to simulate bot trading in section 2.4.3 as the bots will be choosing the top N users by performance to mirror every day.

Additionally, we also have 825,397 mirrorings from June 2011 to November 2013 of traders on other trader, where each mirroring investment consists of the user id of each trader and the user id of the trader being invested in (referred to as the 'parent' trader).

From the mirroring connections (investments), we build a dynamical graph $G_{t,etoro}$ where the nodes are the traders active at time t and the edges are all mirroring connections which were opened before t and closed after t. The dynamical graph changes substantially over time. As we can see in figure 2-3A the number of users in

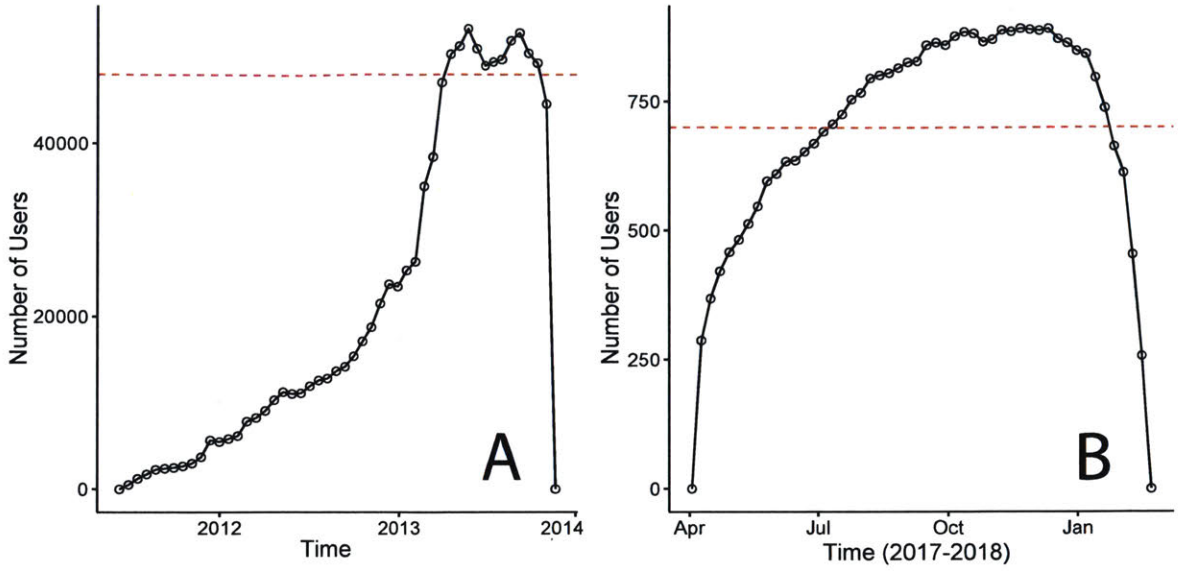


Figure 2-3: Number of users in the eToro (**A**) and Darwinex (**B**) dynamical graph G_t at each time. As we can see, after an initial growth period, the number of users stabilizes around 50,000. In our analysis we take that period of time as our observation period Ω.

$G_{t,etoro}$ increases steadily from the beginning of the platform and it reaches a stable period at the end of 2013. The drastic decrease at the end is an artifact of the mirroring connections ending after the window of available data. The first part of the growth is due to users' coming to the platform becomes it becomes popular. We want to understand user's behavior once the platform is stable as we do not want network growth dynamics to influence our analysis: we are only interested in user's rewiring – mirroring – dynamics and their impact on performance. We will therefore consider only the stable period of time Ω from April 2013 to October 2013 where the number of users fluctuates around 50 thousand. Figure 2-4 shows a snapshot of $G_{t,etoro}$ at the end of the observation period $t = T$

2.3.2 Darwinex Dataset

As in eToro, each trader can buy conventional assets, but can also choose to trade other traders. However instead of copying the trades of other traders, the traders on Darwinex are securitized into their own asset class with dynamic prices set by the platform. For example, a very profitable trader will have a quote price of $1,000

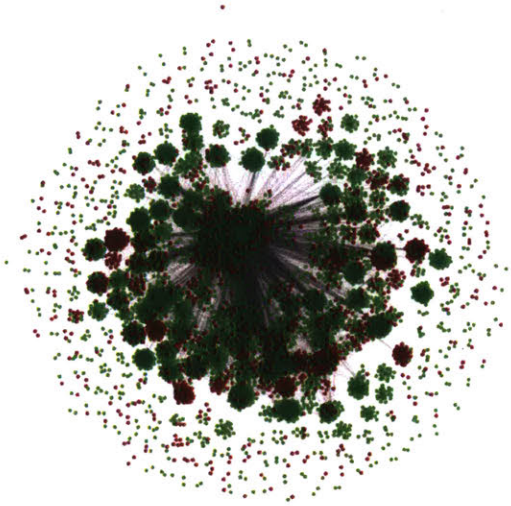


Figure 2-4: Graph of mirroring connections at the end of the observation period Ω. Each node represents a trader in the eToro platform and each links represents mirroring connection between users. Colors indicate positive (green) and negative (red) average trading performance of the users along Ω.

today and, after a series of unprofitable trades, her quote price will decrease to \$800. Traders can therefore choose to buy and sell shares of other traders. There is a 20% fee on profits generated from a mirroring that goes from the mirroring trader to the parent trader.

We have 95,902 *social* trades (mirroring) from April 2017 to Feb 2018 from more than 800 users where each trade consists of the user id of the trader and the parent trader (the trader being invested in), the opening and closing dates, the volume, open and close price and profit received from this trade, amongst other information.

Similar to eToro, we build a dynamical graph $G_{t,darwin}$ where the nodes are the traders active at time t and the edges are all mirroring connections which were opened before t and closed after t. Again, the dynamical graph changes substantially over time. As we can see in figure 2-3B the number of users in $G_{t,darwin}$ increases steadily from the beginning of the platform and it reaches a stable period in the middle of 2017. As in eToro, since we want to understand user's behavior once the platform is stable, we will consider only the period of time Ω from July 2017 to December 2017 where the number of users fluctuates around 650.

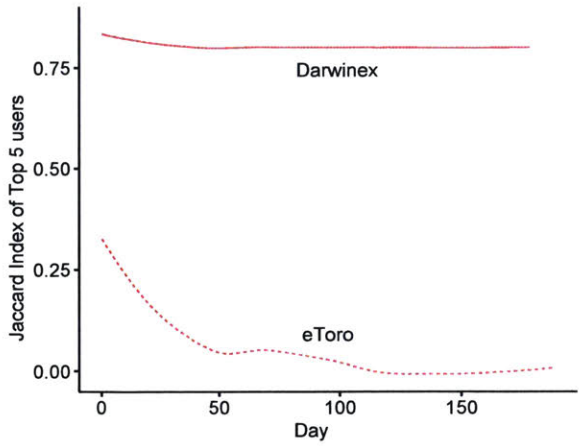


Figure 2-5

Calculating the ROI of each trader over a time period T is done by computing the ROI of the quote price Q_t of a trader i over the time period (after accounting for the 20% fee) as $R_{i,T} = \frac{0.8\times(Q_t - Q_{t-T})}{Q_t}$. We verified with Darwinex that this how ROI is calculated.

2.3.3 Difference between platforms

Darwinex and eToro differ fundamentally in one aspect: the top traders (ranked by ROI) on Darwinex are consistently the same people (as confirmed by the Darwinex themselves) while the top traders on eToro are very different over time, as shown in Figure 2-5. The Jaccard index is defined $J_{t,N} = \frac{|S_0 \cap S_t|}{|S_0 \cup S_t|}$, where S_0 is the set of top N users (here we choose $N = 5$) at the beginning ($t = 0$) of our dataset time period Ω, and S_t is the set of top N users at time t. This might be because of the different revenue structure of each platform: eToro makes money off only spreads, while Darwinex primarily makes money by investing and mirroring their top traders. Therefore, Darwinex has a strong incentive to maintain and encourage traders with top performance. From the perspective of traders in Darwinex, they are also incentivize to be consistent in their ROI as they make money when others follow them (they get 20% of their follower's profit). We will find that this asymmetry leads to different bot mirroring behaviors in section 2.4.3.

2.4 Results

2.4.1 Capacity Limits of Mirroring

To understand users' mirroring strategies we define the following variables of activity and social capacity:

- *Mirroring activity:* $n_i^+(t)$ is the (cumulative) number of mirroring connections opened by user i in the observation period Ω up to time t
- *Mirroring activity:* $n_i^-(t)$ is the (cumulative) number of mirroring connections closed by user i in Ω up to time t
- *Social capacity:* $\kappa_i(t)$ is the number of **still** opened mirroring connections that user i has at time t.

Figure 2-6 shows the dynamics of these variables for a single randomly chosen user in eToro. As we can see the number of opened connections and closed connections increases almost linearly in time as was found in social settings [73], i.e. $n_i^{\{+,-\}}(t) = w_i^{\{+,-\}}t$ (where w_i is a constant). But not only that: we find also that the rate at which connections are created and destroyed is (statistically) the same, that is, that specific user manage his/her mirroring connections in a way so that $\kappa_i(t)$ the number of **still** open connections at time t almost remains constant (black line in figure 2-6).

We find that the behavior we observe for a single user generalizers to all users on both platforms: in figure 2-7A we can see that (on average) the rate of created and destroyed connections for each user is the same, so each user has almost a constant capacity $\kappa_i(t) \sim \kappa_i$. The distribution of capacities for each platform is shown in Figure 2-8A.

Thus each user's strategy can be describe by two quantities: their almost constant social capacity κ_i and his/her activity $n_i^+ = n_i^+(T)$ [or $n_i^-(T)$]. The activity and capacity of each users are not very correlated (see figure 2-7B). We have users that have a very small capacity (e.g. 5 open mirroring connections at any time) but they change very quickly those connections (e.g. having an activity of 200 connections)

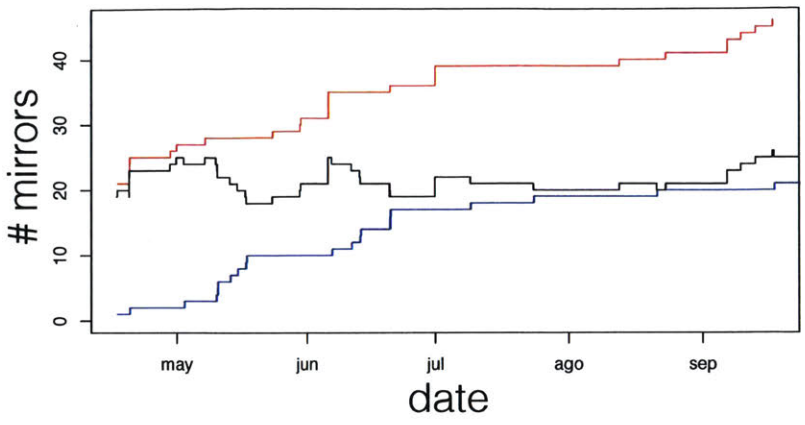


Figure 2-6: Evolution of the number of added, removed and overall mirrorings of a random user. Red line shows the cumulative number of opened mirrors as a function of time shifted by the number of open connections at the beginning of Ω, i.e. it is $\kappa_i(0)+n_i^+(t)$. Blue line shows the cumulative number of closed connections $n_i^-(t)$, while the black line shows the instantaneously opened connections at time t A) Correlation between the number of created mirroring relations n_i^+ and the number of relations destroyed n_i^- for each trader during our observation period Ω. The red line correspond to the $y = x$ line. B) Relationship between the mirroring capacity for each user κ_i and his/her activity, i.e., the number of created links. The red line shows the $y = x$ line.

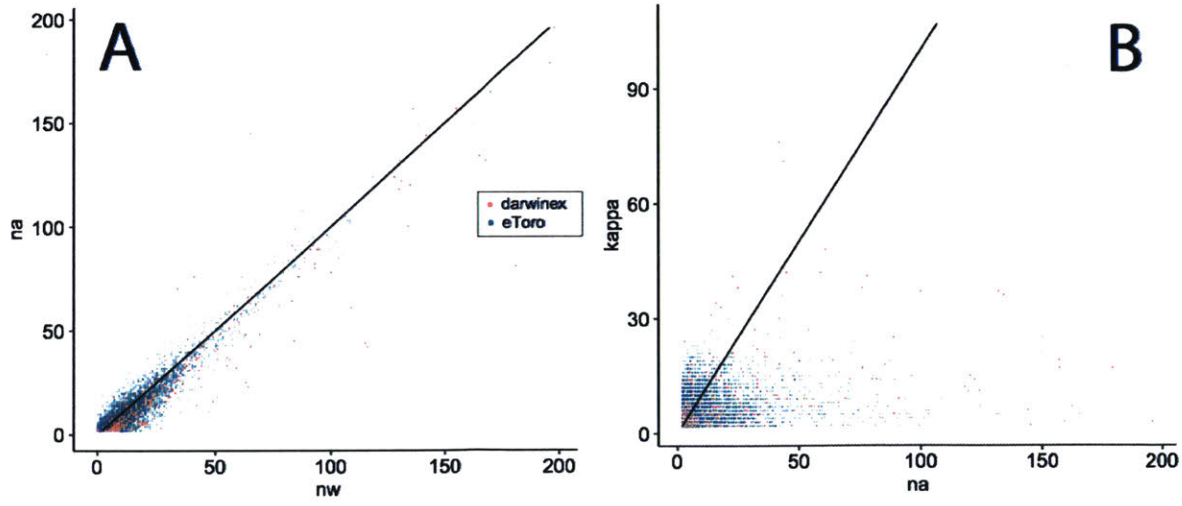


Figure 2-7: **A**: Relationship between the number of created mirroring relations $n_i^+(T)$ and the number of relations destroyed $n_i^-(T)$ for each trader in the observation period. The lines correspond to $y = x$. **B**: Relationship between the mirroring capacity for each user κ_i and his/her activity, i.e., the number of created links $n_i^+(t)$. The line shows $y = x$.

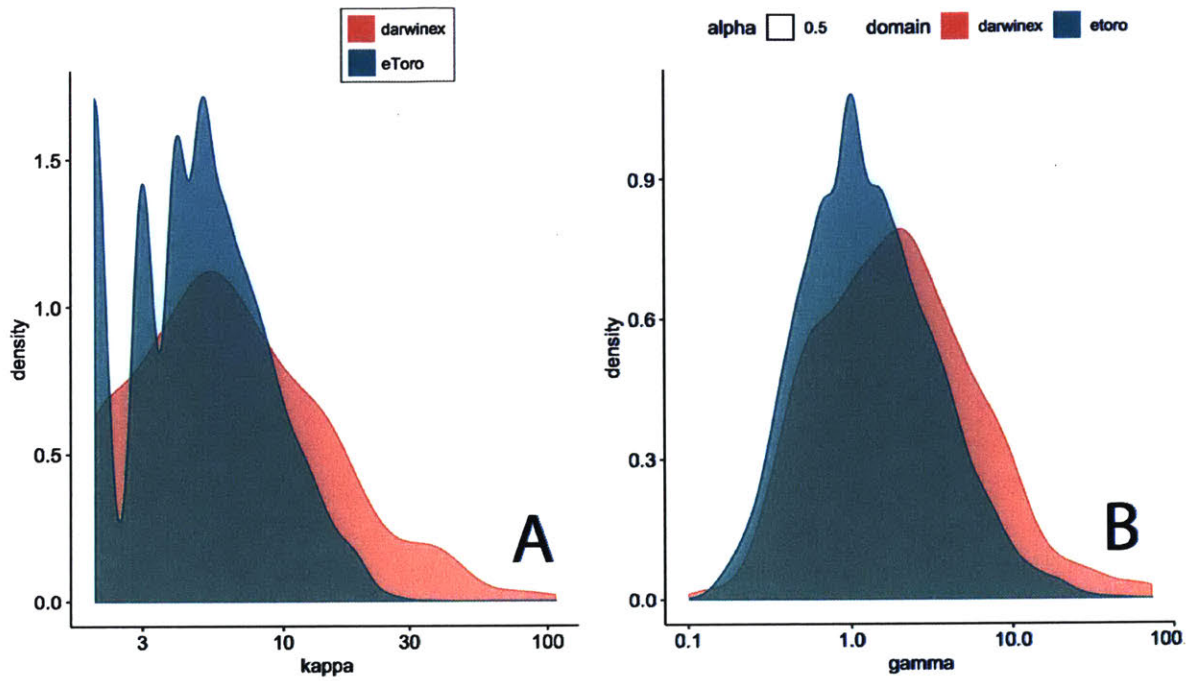


Figure 2-8: **A**: The distribution of capacities κ for each platform. **B**: The distribution of exploration strategies γ for each platform.

which produces a very large and fast turnover of their mirroring connections. We will call those users *social explorers* as in [73]. On the other hand we have users with large capacity (20-30 connections opened at any time) and very small activity (e.g. having opened/closed only 3 mirroring connections). Once again as in [73] we call these users *social keepers*. The distribution of gamma for each platform is shown in Figure 2-8B.

To quantify the strategy we define the variable $\gamma_i = n_i^+/\kappa_i$, where

- $\gamma < 1$ means that the strategy followed by user i is that of a social explorer,
- $\gamma > 1$ correspond to the social keeper strategy.

2.4.2 Performance and Mirroring

Having established that there are capacity constraints on trader's mirroring behavior, the next question is whether these constraints have any impact on the performance of users. Since, in financial markets, it is always good to exploit new information or

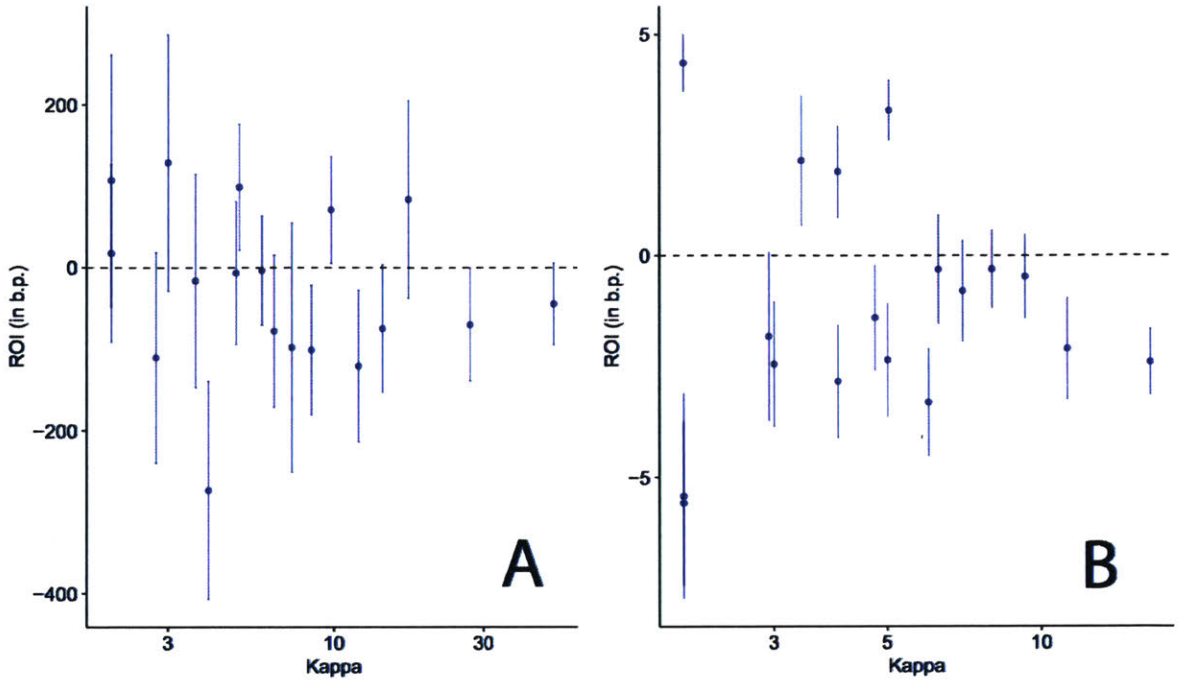


Figure 2-9: Average performance [measured as the average ROI basic percentage points] as a function of κ_i for Darwinex **(A)** and eToro **(B)**. Surprisingly, this shows that there does not seem to be a discernible correlation between capacity and ROI: having higher cognitive capacities does not lead to higher performance!

patterns, one might guess that higher performance would come from higher capacities κ_i.

Using the measures of ROI previously described, we can calculate the average performance $R_{i,\Omega}$ of a user i over time period Ω, and see if there is any relation to κ_i. We choose only users with at least one mirroring link over the whole time period of the data set (i.e. $n_i > 1$ as otherwise users have been inactive), resulting in 6,320 users in eToro and 300 users in Darwinex. We then bin users in 18 bins with the same number of users each based on their κ_i, and report the average ROI of users in each bin.

As can be seen in Figures 2-9A and B, there does not seem to be any discernible trend between a user's capacity for mirroring links with others and the user's performance in either platforms.

We can repeat the previous investigation (with the same binning over γ_i) but using a user's exploration strategy γ_i instead of the capacity κ_i. As can be seen in figures

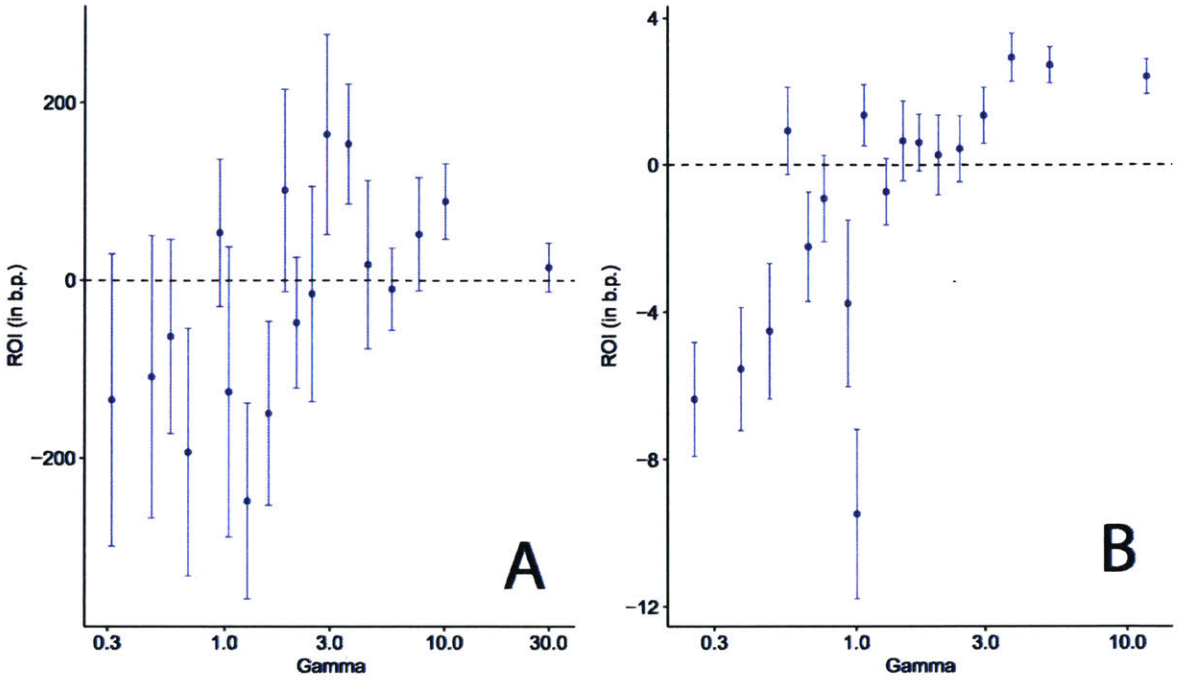


Figure 2-10: Average performance [measured as the average ROI basic percentage points] as a function of γ for Darwinex **(A)** and eToro **(B)**. Users with higher γ have higher performances. The transition from negative to positive ROI is around 1.0 (keepers are defined as $\gamma < 1$, while explorers have $\gamma > 1$). Therefore, the more of an explorer a trader is, they better they do.

2-10A and B, the ROI of users significantly increases with γ. This means that social explorers have on average better performance than social keepers: in both platforms, users who are constantly exploiting new sources of information (i.e. better traders) are better off that those users that stick to a particular set of mirroring connections for a large period of time, irrespective of their social capacities κ_i.

2.4.3 Beyond Human Hypothesis

The average exploration strategy γ of humans on Darwinex is 2.22 and 0.59 on eToro. Given that we observe a strong correlation between a user's exploration strategy γ_i and their ROI, we hypothesize that humans with even higher exploration strategies γ_i would lead to even higher performances.

To test this hypothesis, we build bots with unlimited capacities for mirroring and simulate their strategies on the data. Because we are focused on only investigating if

expanding the capacity of the bot beyond that of humans increased performance, and not the impact of more sophisticated exploration strategies than what humans use (as described by [63]), we only need to test a very simple strategy: for every refresh time interval τ (e.g. every day), at time t (the beginning of an interval), the bot initiates a new mirroring link with each user in the set of top N_t users (where $N_t = \kappa_{bot}$), and invests \$1 equally among these N_t users.

At the end of the next interval $t + \tau$, the bot ends it's mirroring connection with each user in N_t and calculates the ROI obtained during this interval as $r^{bot}_{t,\tau,\kappa} = \sum_{i=1}^{N_t} r_{i,t+\tau}$, where $r_{i,t+\tau}$ is the ROI of each user in N_t at the end of interval $t + \tau$. If a user has no ROI during interval $t + \tau$, their ROI is treated as zero. The length of τ ranges from 1 day to 75 days. The maximum value of τ is 75 to allow for at least 2 intervals during the period of data we have for each platform, approximately 180 days. The average mirroring link refresh time for humans, τ, on Darwinex is 10.2 days and 11.0 days on eToro. We tested our bots over a range of values of τ to add stochasticity to our results for more robust hypothesis testing. n_i and γ for bots was defined similarly to humans.

Given this simple strategy, we can only control the value of κ and τ for each bot and not directly the exploration strategy γ, which is a function $\gamma(\kappa, \tau)$ of κ and τ. By controlling κ and τ, each over a range of 75, we end up with 75^2 unique bots strategies (75 different values of τ and we test capacities κ up to 75). Since the exploration strategy function $\gamma(\kappa, \tau)$ is non-linear, two bots of different κ and τ might end up with similar γ values. As shown in Figure 2-11A, using this simple strategy, the values of γ achieved by the bots can be larger than those of humans in eToro.

We calculate the ROI of a bot, $R^{bot}_{\kappa,\tau}$, by averaging the ROI over each time interval t of the bot $r^{bot}_{t,\tau,\kappa}$ from the first time interval up to Ω/τ: $R^{bot}_{\kappa,\tau} = \frac{1}{\Omega/\tau} \sum_{t=1}^{\Omega/\tau} r^{bot}_{t,\tau,\kappa}$. We then calculate the average ROI $R^{bot}_{\kappa,\tau}$ of all bots by their resulting exploration strategy $\gamma(\kappa, \tau)$. As shown in Figure 2-12A, bots with higher exploration strategies $\gamma(\kappa, \tau)$ can outperform humans significantly in eToro, validating our hypothesis that higher exploration leads to higher performance.

However, we do not find the same behavior on Darwinex. As shown in Figure

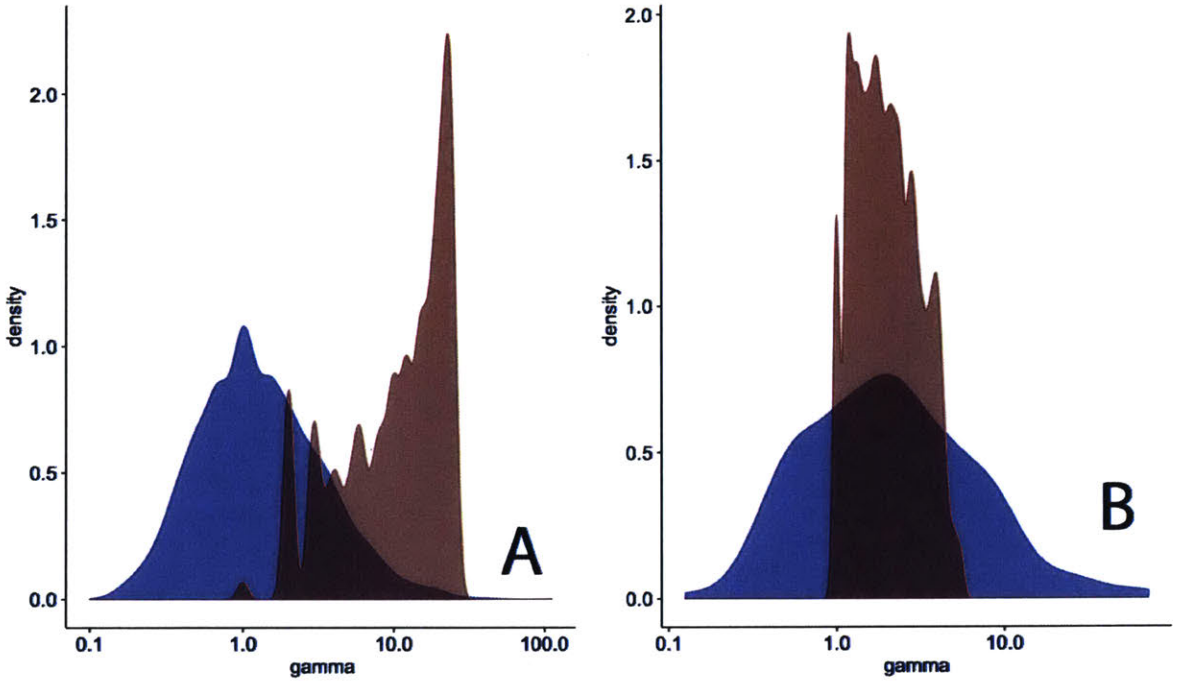


Figure 2-11: Our bots can achieve higher exploration strategies γ than humans in eToro (**A**) because the top people being followed through our simple strategies are always changing, whereas they do worse in Darwinex (**B**) due to the consistency of the top traders in Darwinex.

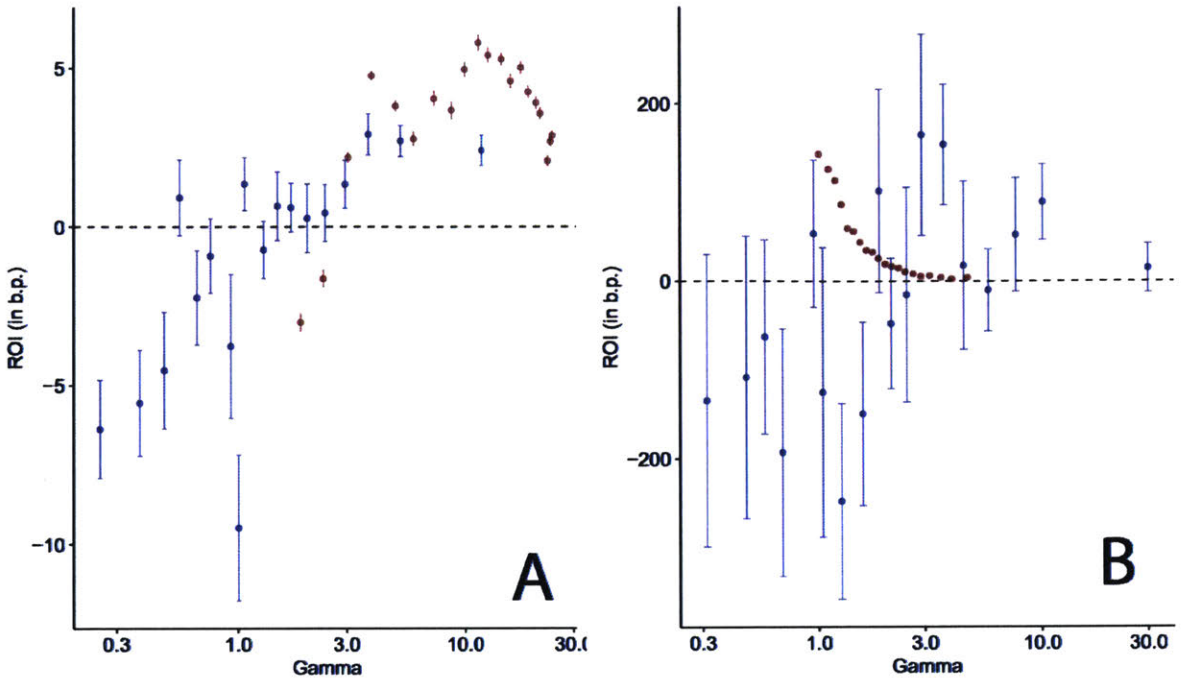


Figure 2-12: Our bots can achieve higher ROI in than humans in eToro (**A**) due to higher exploration strategies γ , whereas they do worse in Darwinex (**B**) due to the inability of our simple mirroring strategy to achieve higher γ.

2-11B, our bots achieve smaller $\gamma(\kappa, \tau)$ on Darwinex with our simple bot strategies. This is because, given the fundamental difference described in section 2.3.3, sticking to a bot strategy which chooses the top users on Darwinex for every time interval results in very little exploration γ because the same users are being picked over and over. Consequently, our bot's performance is quite different on Darwinex than on eToro, as shown in Figure 2-12B: not only are bots not better than humans, but higher bot exploration results in lower ROI. Overall, this means that in a consistently high-performing market, exploitation (choosing the best users) is better than exploration. Keepers do better when exploration causes high regret, a finding consistent with the Multi-Armed Bandit literature[20].

Therefore, when using a simple trading strategy, it is better to be an explorer in an uncertain platform (eToro) as exploration is rewarded, and to be a keeper in a consistent higher return platform (Darwinex).

2.5 Contribution

In this chapter, we observe that even financial traders – whose performance depends on their ability to learn new strategies from from others – exhibit cognitive constraints on the number of social connections that they have. Interestingly, their performance is not correlated with their congnitive limit, but is however strongly predicted by their amount of exploration. We also build bots that transcend human congnitive limits, we show that they can outperform humans even when using very simple bot trading strategies.

Chapter 3

Modeling Social Learning

3.1 Purpose

Given that people exhibit strong cognitive constraints in how they learn from social information, the question is how do such constraints mold the way people learn socially? Through a novel study of unprecedented scale, we model people's belief update process after being exposed to dynamic social information and price history using Bayesian models of cognition. We observe many inductive biases, such as the fact that people make strong distributional assumptions on the distribution of the belief of their peers, and that they prefer to learn from social information than from non-social data.

3.2 Background

In order to shed some light into this question, we collected data through a large online experiment where 2,037 mid-career financial professionals made a total of 9,268 estimates of future financial asset prices (the S&P 500, WTI Oil and gold prices) in a series of seven independent live prediction rounds. By carefully instrumenting the data that each individual is shown after they made their initial pre-exposure prediction of the asset prices, and then recording their revised prediction, we are able to investigate the inductive biases present in people's belief update process by using

formalism inspired by Bayesian models of cognition.

This question has been studied in the context of the Wisdom of the Crowd (Woc)[42, 106] where individuals are asked to make a predictions of a certain quantity, such as the number of beans in a jar, and their average predictions are compared before and after exposure to certain information. If, for example, this information is the predictions of others in the crowd, this procedure would allow us to observe if exposure to this information improves or worsen the collective accuracy of the crowd.

Although it has been established that the average prediction of the crowd can be quite accurate in a range of domains (e.g., predicting the reproducibility of scientific research [33], guessing the weight of a bull [42], estimating caloric content of food [14], prediction markets [4], predicting stock market prices [85]), there is still disagreement in the literature as to the role of information exposure on collective accuracy.

Past work has shown that learning from peer information can decrease the diversity of opinions even if it can improve collective accuracy[68]. Similarly, the confidence levels of individuals in their prediction has been shown to influence others to change their predictions for the worse [78]. In the domain of online ratings, prior ratings have been observed to bias individual ratings asymmetrically based on the positiveness and negativeness of prior ratings. It has also been shown theoretically that the crowd might not converge to the true estimate if the influence of the most influential person does not vanish as the size of the crowd grow [110]. If individuals have different observational sensitivities, they will be collectively less accurate [7].

This is because conflicting effects are at play. One one hand, people may learn from each other and become more accurate. One the other hand, when people are exposed to similar information (such as the predictions of their peers), correlation between people's predictions can creep in or users might get influenced by their peers, leading to a biased collective estimate. In this work, we design a novel online experiment where these effects can be studied with precision and at an unprecedented scale.

There has been growing work in recent years trying to understand how to make the Wisdom of the Crowd more accurate under information exposure settings. For example, there is work on improving the network topology of information sharing[86,

14], optimally allocating crowdsourcing tasks to individuals [58] and recalibrating forecasts against systematic biases of individuals [110].

However, to the extent of the authors' knowledge, no work has attempted to improve the Wisdom of the Crowd accuracy based on understanding the delicate statistical processes individuals use to update their belief using probabilistic models from cognitive science[47]. This is the novel approach that we take.

Specifically, we hypothesize that what information sources individuals use to update their belief - and the processes they use to update their belief - might provide useful signals of accuracy. By selecting these individuals who are more accurate than their peers, we can obtain a more accurate collective prediction (i.e. we obtain a 'smarter' wisdom of the crowd). Previous work has shown that selecting users can lead to improvements (such as by finding users who are resistant to social influence[70]), but no prior work has specifically investigated the belief update process of individuals.

Overall, there has been limited investigation of the statistical nature of individual and collective estimation in the Wisdom of the Crowd even though there is evidence that individuals do not always give their best guess when answering a question [114], but instead sample from an internal distribution. Additionally, there is strong justification that humans employ Bayesian processes when the right prior and likelihood distributions (evidence) are used [48], and that they can even be Bayes optimal [7]. Finally, there is evidence that specific brain regions are responsible for perceiving social information [54] and in reasoning about others' minds [97].

3.3 Experimental Design

2,037 mid-career financial professionals were recruited as part of an advanced financial online course where, as homework, they were asked to make predictions of prices of financial assets (e.g. S&P 500 and gold prices) during seven separate consecutive 3-week rounds over the span of 6 months, resulting in 9,268 predictions. We are releasing the data collected in this study. The individuals consented to their data to be used in this study and we obtained prior IRB approval.

During each round, all users made a prediction of the same asset's closing price for the final day of the round. We use the round's last day's closing market price as our measure of ground truth. We chose the start and end dates of each round such that we could use the futures underlying each asset as a measure of the global market's prediction of the price asset. We chose the end date of the round such that the futures price will not be affected by volatility towards its expiration date. Whenever we predict a final closing price, we only use user prediction data up to the week before the day of prediction (i.e., we don't use any data during the last week of the round) so that our predictions are not too easy. One of our rounds of prediction happened to end the day of the Brexit vote, which means that we have prediction data during a particularly tumultuous market period [87]. Our financial data is obtained through Barchart.com's data API.

As shown in Table 3.1, we observe that the crowd predictions are generally accurate as the average group prediction error is much less than the overall price change of the asset for the 3-week prediction period. Additionally, the collective prediction over each round tracks (and sometimes outperforms) the futures of each asset being predicted (we calculate the futures error as the difference between the futures price and the asset price). We also find that the crowd is generally doing more than just linear extrapolation in time. A screenshot of the user-interface is shown in Fig 3-1.

To model the belief update process of individuals (illustrated in figure 3-1), we designed the data collection process as follows: every time a user makes a prediction of an asset's future price through our platform, the following prediction set is collected: B_{pre}, B_H, B_T and B_{post} is collected:

- A "pre-exposure" belief prediction B_{pre} which is independent of any social information. For example, a user might show up on the platform and predict that the closing price of the S&P 500 to be $2,001 on June 24 2016.

- The list of prior predictions B_H up to this point made my other users is used to create the histogram of peer predictions (the "social information") which is shown to users in the form of a pop-up. Additionally, we display the 6-month

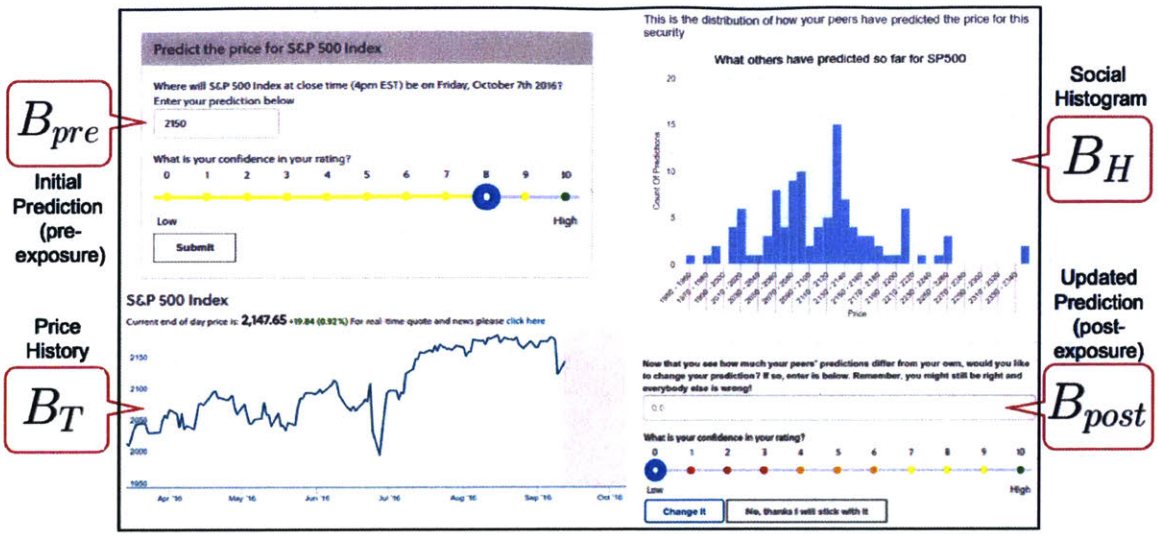


Figure 3-1: An annotated screenshot of how our data is collected: the pre-exposure prediction B_{pre}, the social histogram B_H, the price history B_T and the updated prediction B_{post}. The final ground truth of the asset's closing price will be V (not shown here, realized at the end of the round).

of history of the asset's price B_T up to this point.

- The revised "post-exposure" prediction B_{post}. For example, after seeing the social histogram and asset price history, a user might update their belief to be \$2,201. Since the real price (the groud truth) ended up bring \$2,037.41, this user became more accurate after information exposure.

The ground truth of each round, V, the closing price of the asset on the last day of the round, will be compared to B_{pre} and B_{post} to understand the effect of information exposure on accuracy. Other information is also collected such as confidence ratings, demographic data, survey questions and course grades but we do not use this data in this work.

We collected 4,634 such prediction sets (resulting in 9,268 predictions, one pre-exposure and one post-exposure) across our seven prediction rounds. We make sure that the "pre-exposure" prediction is made before any social information is shown. We present a unique histogram for every new prediction (as it is built using past predictions up to this point), as well as a unique price history time series (as it shows the 6-month price data up to the time of prediction), and we require all users to make

	Round						
	1	2	3	4	5	6	7
Asset	**SP500**	**WTI Oil**	**Gold**	**SP500**	**SP500**	**SP500**	**SP500**
Grounth Truth ($)	2037.41	45.95	1335.80	2153.74	2126.41	2191.95	2262.53
Num. Prediction Sets	284	207	134	1174	925	1441	469
Price Change (%)	4.01	11.03	3.63	1.77	1.75	2.24	3.56
Momentum Error (%)	6.66	16.4	1.26	1.62	2.75	0.75	3.10
Crowd Mean Error (%)	2.22	4.95	0.46	0.84	0.58	3.20	2.40
Futures Mean Error (%)	2.03	3.05	0.94	0.38	0.40	0.48	1.50

Table 3.1: Summary of data collected. Our crowd is accurate, and sometimes even outperforms the futures underlying the asset. Predictions made by users are more accurate than simple linear extrapolation.

a post-exposure prediction.

3.3.1 Effect of Information Exposure

We can calculate the average B_{pre} and B_{post} for each round, and compare their error relative to the ground truth. This would allow us to observe whether exposure to information across all seven rounds for all 4,634 unique prediction sets leads to increased or decreased accuracy. As shown in figure 3-2, the aggregate error of the crowd does not get significantly better or worse after exposure to social information. This could be due to various factors such as the much larger size of our dataset compared to previous studies (number of users N=144 in [68], N=59 in [78] compared to N=2,037 in our case), or the fact that we are predicting over harder problems in our studies that even us experimenters do not know the ground truth until the closing day's price is realized weeks later (previous studies focused on estimating the number of beans in a jar[14] for example).

Note that errors are higher for round 2 (WTI Oil asset price) only because the price of oil is very small (about $45) compared to the S&P 500 price (about $2100) such that relative errors for the same average dollar price error made by individuals seem larger.

However, it might still be possible to recover individuals whose post-exposure prediction make them more accurate, and we hypothesize that what information sources they use to update their belief might be a useful signal of accuracy.

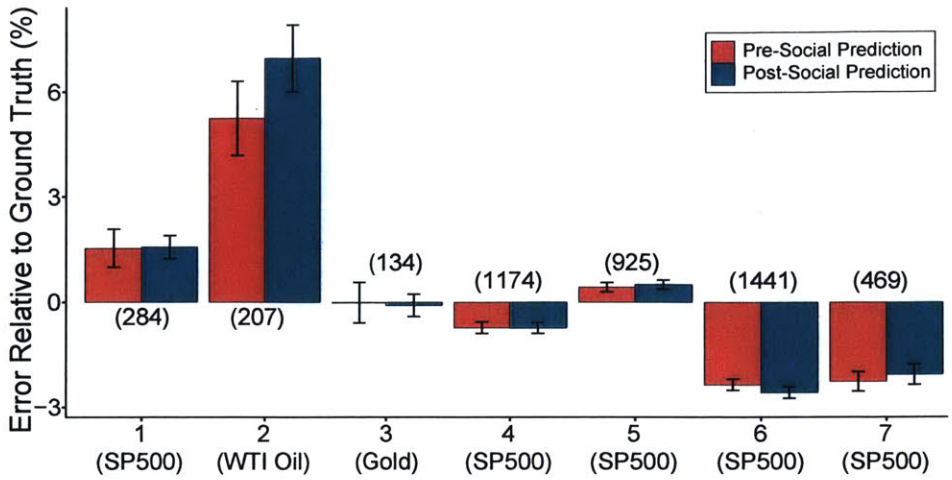


Figure 3-2: Aggregate error of pre-exposure and post-exposure predictions (relative to ground truth) across all seven rounds for all 6436 unique prediction sets (numbers in parenthesis are the number of prediction sets for each round). Negative errors means the crowd underestimated relative to the ground truth.

3.4 Modeling Belief Update

3.4.1 Formalism from Bayesian Models of Cognition

First, we have to understand how people update their belief (prediction of the asset price). Using formalism inspired by Bayesian models of cognition [47], we can model the 4,634 prediction sets collected over many rounds, at a high level, as a Bayesian update: $posterior \propto likelihood \times prior$. To use this formalism, we need to choose a prior for each individual's belief, and the likelihood distribution (evidence) an individual is updating with. Previous work has shown that choosing even the simplest distributional form for the prior and likelihood (e.g., Gaussian likelihood and priors) can lead to extremely accurate posterior predictions of people's estimate of domains ranging from predicting human lifespans and movie grosses[48].

Here, we describe at a high level this Bayesian update, and include the model derivations and the details of the computation of the posterior distribution in the next section. As we are interested in how individuals update their belief of the asset's future price (ground truth) V based on the information we expose them to, the choice of how to approximate the prior distribution is straightforward: $P_{prior} \approx P(B_{pre})$, the

distribution over an individual's pre-exposure beliefs of V. Since we obtain only one sample, B_{pre}, for each user and cannot observe the full distribution $P(B_{pre})$, we will discuss how, when needed, we estimate $P(B_{pre})$ in the next section.

There are two main candidates for which likelihood (evidence) users might employ: using the assets' price history B_T users are shown, giving us $P_{likelihood} \approx P(B_T)$, or analogously, the social histogram B_H, giving us $P_{likelihood} \approx P(B_H)$. The *modeled* posterior prediction $P_{posterior}$ can therefore be approximated as $P_{posterior} \propto P(B_H) \cdot P(B_{pre})$ in the case of the social histogram, and $P_{posterior} \propto P(B_T) \cdot P(B_{pre})$ when users learn from the past price history. Users might be using other sources of information (such as financial news) but, as per the efficient market hypothesis[71, 39], we can assume that all other information sources about the asset price that could be used as the likelihood distribution is factored into the asset price history[71, 39].

We do not make any other assumptions in terms of what data to use to approximate the likelihood and prior distributions. Our assumptions are based on previous work focused on how people learn using different likelihood and priors[48], and, here, we only extend it to the case when people learn from social data and price data given our experimental setup.

We are interested in the residuals between the *modeled* posterior prediction $P_{posterior}$ and the real collected posterior distribution $P(B_{post})$. The question is then how to exactly compute the posterior distribution. For example, should we assume a certain parametric form of the distribution (e.g. normally-distributed), or should we numerically (using Monte Carlo methods) compute it?

3.4.2 Derivation of Parametric Model

Our goal was not to search the space of possible belief update models and find the best model. We are interested in comparing the simple, theoretically-motivated and robust models inspired from previous work from Bayesian cognition that still give us insights into the kind of information people use to update their belief, and that will allow us to select predictions for higher Wisdom of the Crowd accuracy. We leave to future work the investigation of the best models of social learning, including

those using data that we did not investigate in this datase (e.g. confidence ratings of predictions, or level of education and skill of users).

We focus on the derivation of the parametric Gaussian models `GaussianSocial` and `GaussianPrice` in this derivation. We describe `GaussianSocial` here. `GaussianPrice` follows the same derivation, substituting the social historgram B_H with the price history B_T.

Because `GaussianSocial` involves a simple linear average, there are many ways to derive this belief update. We present a derivation based on the Bayesian cognition literature since this literature was the inspiration for our use of this model. Our notation follows that of Kim et al. [59].

We suppose that people think each asset has a true value, V^*, which people are trying to estimate to predict the future asset value, V (the ground truth); that prior beliefs about V^* follow a Normal prior distribution, $V^* \sim Normal(\mu_{prior}, \sigma_{prior})$; and that evidence about V^* can be understood as being generated from a Normal distribution, $Normal(V^*, \sigma_{data})$. In this case the posterior beliefs people have follows a simple form. Letting information content be defined as the inverse of the Normal distribution's variance $I = \frac{1}{\sigma}$, we have that

$$\mu_{posterior} = \frac{\mu_{prior} \cdot I_{prior} + \mu_{data} \cdot I_{data}}{I_{prior} + I_{data}}.$$

Supposing further that B_{pre} represents each person's prior mean, and that the social histogram is treated as representing the information content of data about V^*, then we have:

$$\mu_{posterior} = \frac{B_{pre} \cdot I_{prior} + \overline{B_H} \cdot I_{data}}{I_{prior} + I_{data}}.$$

The `GaussianSocial` rule therefore can be viewed as reflecting an assumption of a Normal distribution as a mental model, and assuming private information and social information have the same information content ($I_{prior} + I_{data}$), which gives:

$$\mu_{posterior} = \frac{B_{pre} + \overline{B_H}}{2}.$$

3.4.3 Numerical Models

In the numerical (non-parametric) models, we bin the likelihood distributions and numerically calculate the posterior distribution. Because we do not have access to the actual distribution of the prior belief of each individual (as we only have an individual point estimate for each prediction set), we have to assume a parametric distributional form for the prior distribution. Since we are not searching for the best possible model that explains fully the belief update process of people but instead want to gather insights into how people update their belief, we model the prior to be Gaussian, with the mean set as the pre-exposure prediction of an individual, B_{pre}, and the standard deviation set to be the standard deviation of the social histogram B_H or the standard deviation of the price history B_T, , depending on likelihood used.

We calculate the posterior distribution $P_{posterior}(b)$ of an individual's post-exposure prediction b in the following way: let b_j be a unique value in $\mathbf{B_H}$, and $P_{B_H}(b_h)$ be the probability density of b_h in $\mathbf{B_H}$. Let $P_{prior}(b)$ be the density of b in the parametrized prior distribution. The posterior distribution for the numerical model is defined as $P_{posterior}(b) = \frac{P_{B_H}(b) \times P_{prior}(b)}{\sum_{b_j \in B_H} P_{B_H}(b_j) \times P_{prior}(b_j)}$ when using the social information B_H. After computing this posterior distribution, we report the mean of the distribution as the user's predicted updated belief B_{post}.

3.4.4 Momentum Transformation of Price History

From the price history B_T, the daily rate in price change is calculated, and this histogram of price change per day is used to extrapolate and predict asset prices. Specifically, a daily rate, r_t, of asset price change is calculated for each day during the 6 month interval that a user is shown, $r_t = \frac{B_t - B_{t-1}}{B_t}$. These rates are then used to create a histogram (just as before) and this rates histogram is utilized for both the parametric models and the numerical models. In the parametric model `GaussianPrice` for example, the mean of this histogram (which is mean rate $\bar{r}_t$ over the 6 month period) is multiplied by the number of days between the pre-exposure prediction and the final day of the prediction round (for when the asset's price is

being predicted) to obtain the post-exposure prediction, $B_{post}^{pred} = B_{pre} + B_{pre}.\bar{r}_t.n_{days}$. The same calculation is done for the numerical model, but for each bin in the rates histogram (as will be discussed next).

3.4.5 Evaluating Model Error

For the parametric models, we compute the relative residual error between the model's prediction of the posterior, and the actual post-exposure prediction, $(\mu_{\sim P_{posterior}(V)} - B_{post})/B_{post}$. For the numerical model, we compare the mean of the computed posterior $P_{posterior}$ to the actual post-exposure prediction of each prediction set, B_{post}.

For all models, the 95% confidence intervals are calculated as follows: we assume the data follows Student's t-distribution since the variance of the true distribution is unknown and therefore we estimate it from the sample data. Let s_e be the estimated standard error of the sample mean, and t_e be the t value for the 95% confidence interval desired which can be computed via inverse t distribution. The lower and upper limits for the 95% confidence interval are $[\mu_e - t_e s_e, \mu_e + t_e s_e]$, where μ_e is the estimated sample mean.

3.4.6 Model Performances

Our goal is not to search the space of possible belief update models and find the best model. We are interested in comparing the simple, theoretically-motivated and robust models inspired from previous work from Bayesian cognition that still give us insights into the kind of information people use to update their belief, and that will allow us to select predictions for higher Wisdom of the Crowd accuracy.

Focusing first on the case when people are learning from social information, we observe that a parametric model that assumes both the prior and (social histogram) likelihood to be normally distributed outperforms more complex numerical models. As shown in Figure 3-3, the parametric model of how people update their predictions, B_{post}, is $P_{posterior} = (B_{pre} + \overline{B_H})/2$ (we call this model `GaussianSocial`). `GaussianSocial` outperforms the numerical model `NumericalSocial` at predicting

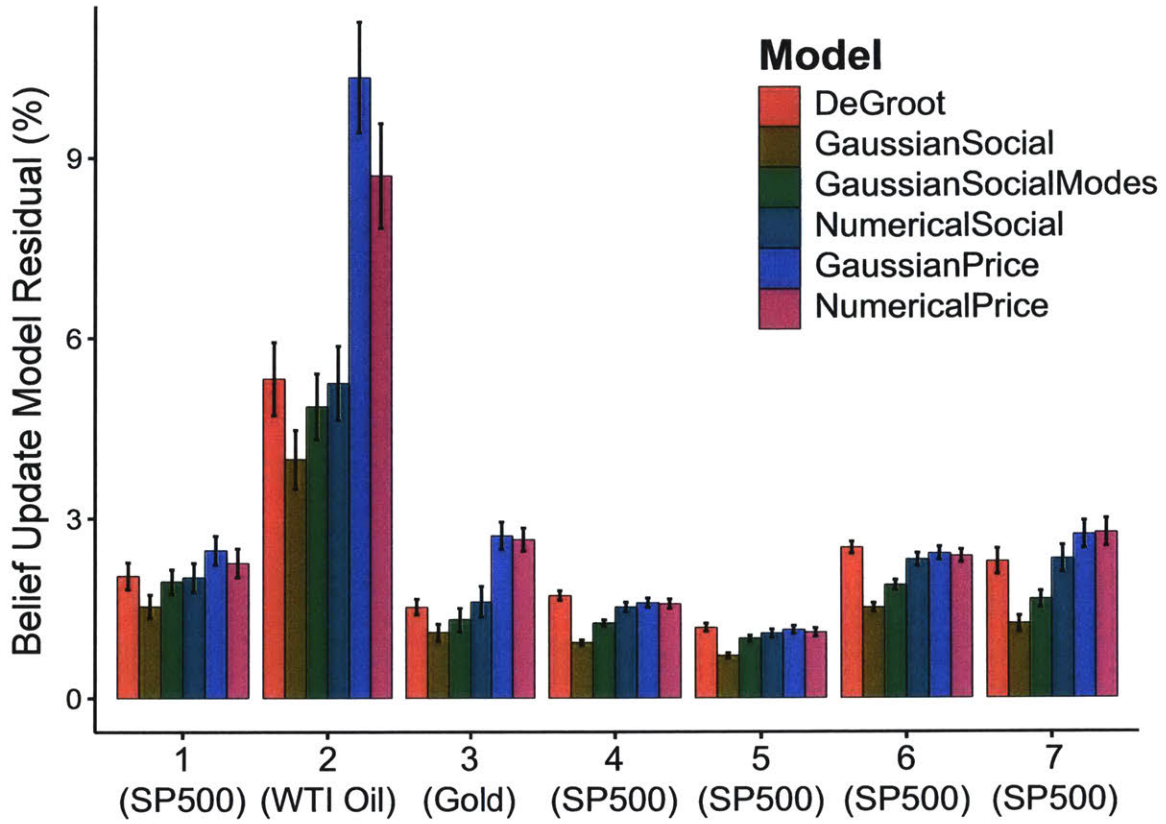


Figure 3-3: Parametric models do better at modeling belief update than numerical models, and models using social histogram as likelihood perform better than models using the price history. Relative error is between modeled post-exposure and observed predictions.

how people update their posterior belief after seeing the social histogram. The y-axis shows the relative residual between the modeled posterior of the model, $P_{posterior}$, and the actual updated predictions of individuals B_{post}. Our parametric model also outperforms the popular DeGroot model[31] commonly used as a benchmark in the literature where an individual update their belief as the weighted average belief of their neighbors.

Additionally, we investigated the case when the social histogram is not assumed to be unimodal but instead is assumed to have several modes. The hypothesis is that perhaps people are drawn to the mode in their peer's beliefs which is closest to their initial prediction, or perhaps they choose the mode that is largest. Using the Hartigan's dip test of unimodality [50], we identified when the social histogram was non-unimodal, and we explored using the largest mode of the distribution using

the `GaussianSocialMultiMode` model but find that the parametric Gaussian model `GaussianSocial` which only assumes one mode still does best, indicating that even when the distribution clearly has more than one mode, people assume the data to be unimodal.

Having investigated the models of belief update when people are learning from social information, we turn to the case when people are learning from the price history. When using the price history B_T, we do not simply bin it but instead use the timeseries to compute a distribution of momentum asset price change because people have been shown to process timeseries using various orders of momentum prediction[91, 83]. We find that the performance of the analogous parametric model $P_{posterior} = (B_{pre}+\overline{B_T})/2$ (we call this model `GaussianPrice`) is indistinguishable from that of the numerical model (`NumericalPrice`) in all but one round. We can add B_{pre} to the average of B_T, $\overline{B_T}$, as the average is also a scalar.

We would expect models that use the price history as the likelihood to have smaller errors as the individuals in our experiment are mid-career financial professionals who would likely believe, as per the efficient market hypothesis, that learning from asset prices is optimal in predicting future prices. However, we observe the opposite: parametric (`GaussianSocial`) and numerical (`NumericalSocial`) models that use the social information as the likelihood distribution to model the individual subjects' post-exposure price prediction B_{post} outperform both parametric `GaussianPrice`) and numerical (`NumericalPrice`) models that use price history.

In summary, we observe that models using social information as the likelihood (evidence) do better than models that use the past price history.

3.5 Improving the Wisdom of the Crowd

3.5.1 Subsetting Predictions based on Information Source

Our hope is that subsetting predictions of users by which information they more likely used to update their belief will allow us to select predictions that are more accurate at

predicting the final price of the asset, the ground truth. These are separate problems: in one case, we are building models of how people are updating their prediction, and measure the performance of these models using the residual is between the actual updated prediction of a user, B_{post} and the modeled updated prediction, $P_{posterior}$, using `GaussianSocial` and `GaussianPrice`. In the other case, we are looking into whether these chosen predictions, B_{post}, are more accurate predictions of the future ground truth. That is, we observe the residual between V and B_{post}. Our hypothesis is that the first residual (of *how* people update their belief) is predictive of the second residual (the *accuracy* of their predictions). This process is illustrated in in Figure 3-4.

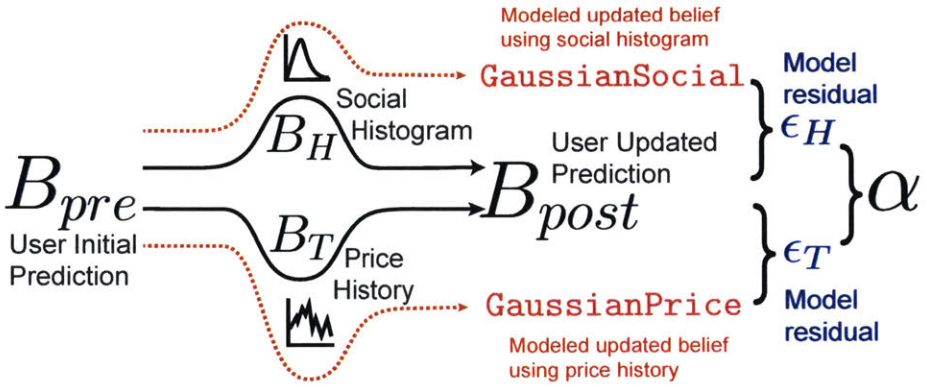


Figure 3-4: For each prediction set, a user might update their belief from the pre-exposure prediction B_{pre} to the updated prediction B_{post} by either learning from social histogram B_H and/or the price history B_T. ϵ_H is the residual between the *modeled* updated prediction `GaussianSocial` and the user's updated prediction B_{post}; ϵ_T is the residual between `GaussianPrice` and B_{post}. α is the difference between ϵ_T and ϵ_H, and will be used to select predictions to compare against the ground truth V.

We can estimate how much each individual used the social information B_H instead of the price history B_T to update their belief, B_{post}, by comparing the residual errors in each case.

To obtain ϵ_H, the model residual when a user is learning from social information, we compare B_{post} to the mean of $P_{posterior}$, as $\epsilon_H = \frac{|\texttt{GaussianSocial} - B_{post}|}{B_{post}}$. Similarly, we obtain ϵ_T, the model residual when a user is learning price history information, by comparing B_{post} to the mean of $P_{posterior}$, as $\epsilon_T = \frac{|\texttt{GaussianPrice} - B_{post}|}{B_{post}}$. We focus only on parametric models to compare data sources for the likelihood distribution

because, as we have observed in the previous section, parametric models are better at predicting belief update in the case of the social information, and indistinguishable from the numerical model for the price history.

By using $\alpha = \epsilon_T - \epsilon_H$, we can estimate how much more accurate the models using the social information B_H instead of the price history B_T were, and use this as a measure of how how likely a user used each source of information to update their prediction. For example, for a prediction set $[B_{pre}, B_H, B_T, B_{post}]$ if $\alpha > 0 \implies \epsilon_T > \epsilon_H$, this means that this prediction set is better modeled using the social histogram of peer's belief B_H as the likelihood (smaller error ϵ_H) instead of the price history B_T. Similarly, when $\alpha = 0$, this means that the model error from using the social histogram is the same as when using the price history, indicating a balanced used of both data sources.

Using α, we can select a subset S_{α_s} of the prediction sets collected such that the α of all predictions set lie in the range $0 \leq \alpha \leq \alpha_s$ (or $\alpha_s \leq \alpha < 0$ when $\alpha < 0$). As an example, if $\alpha_s = 0.3$, this will allow us to select all prediction sets made that are better modeled using the social histogram as likelihood than when using the price history. A still higher value of α_s will lead us to select predictions are even better modeled using the social histogram data than the price history. Using these subsets, we can compare whether predictions selected using either the social histogram or the price history are more or less accurate (compared to the ground truth V). To do so, we select the post-exposure (updated) predictions $\overline{B_{post}^{S_{\alpha_s}}}$ within the subset of predictions S_{α_s} to the mean of post-exposure predictions $\overline{B_{post}^{S_{all}}}$ the unfiltered complete set of predictions S_{all} (unfiltered using $-1 \leq \alpha \leq 1$ as we rescale α to be in the interval [-1,1] for all rounds). We measure the improvement of a subset S_{α_s} as $(\overline{B_{post}^{S_{\alpha_s}}} - \overline{B_{post}^{S_{all}}})/\overline{B_{post}^{S_{all}}}$. This will allow us to investigate if users are more accurate when they pay more attention to the social histogram values B_H than to the price history B_T.

3.5.2 Social Learning as a Signal of Accuracy

We bin the α's from all 4,634 prediction sets into 15 groups of equal size and investigate the improvements of selected groups compared to the whole crowd.

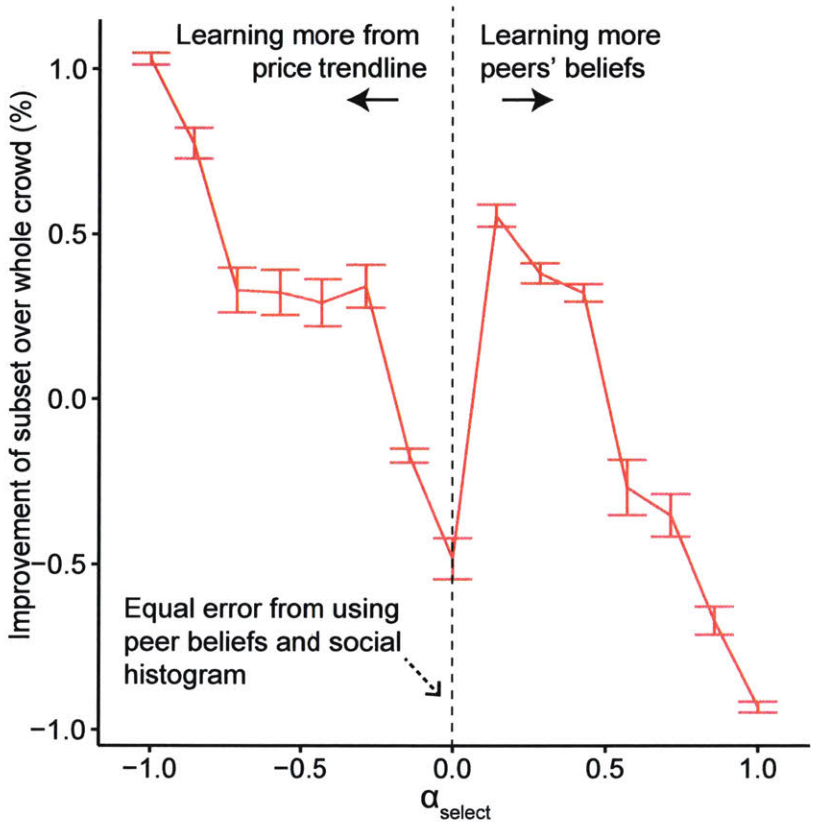


Figure 3-5: Improvement when selecting predictions based on how much more they were likely made use social information ($\alpha > 0$) vs. price history $\alpha < 0$. We see a clear improvement when predictions made using price history are selected, in agreement with the efficient market hypothesis.

As shown in figure 3-5, subsetting over post-exposure (updated) predictions by information source can lead to a significant improvement in accuracy of our subset compared to the whole crowds' post-exposure prediction. Specifically, we observe a maximum improvement of up to 1.03% (statistically significant, 95% confidence interval [1.01, 1.04] using 100 bootstrap draws) when using a subset of predictions selected using $\alpha_s = -1.0$ compared to using the whole crowd S_{all}. The improvement values and their accompanying α_s is included in Table 2 in the appendix.

As per our definition of α, this improvement comes from users who updated their belief strongly based on the price history instead of the social histogram. This result is in accordance with the efficient market hypothesis which suggests that using past prices is the most informative strategy to predict future prices. Conversely, users do tend to update their belief based on social information tend to have increasingly more inaccurate predictions compared to the crowd.

We also calculate the variance in prediction accuracy at various values of α by calculating the standard deviation of the subset of prediction S_{α_s}. We use a Pareto curve[72] to visualize the trade-off between the mean and standard deviation of prediction error by users across our seven rounds, shown in figure 3-6. We observe that although people who rely more on price history are more accurate, there is increased variance in their prediction, i.e. there is a risk-return trade-off between listening to one's peers versus looking at the price history.

3.5.3 Predicting under Uncertainty

We were fortunate that one round happened during the Brexit vote which caused a lot of market uncertainty[87]. We, therefore, have a unique opportunity to investigate if our results hold even during uncertain times.

In all previous results, we took care not to use the last week of data when we make predictions so that the predictions are not too easy. In this case, exceptionally, we use the last week of data because the market shock was during the last week of the round and thus prediction will not be too easy. Crucially, this will also enable us to see if our previous result holds in periods of market uncertainty. Therefore, we run

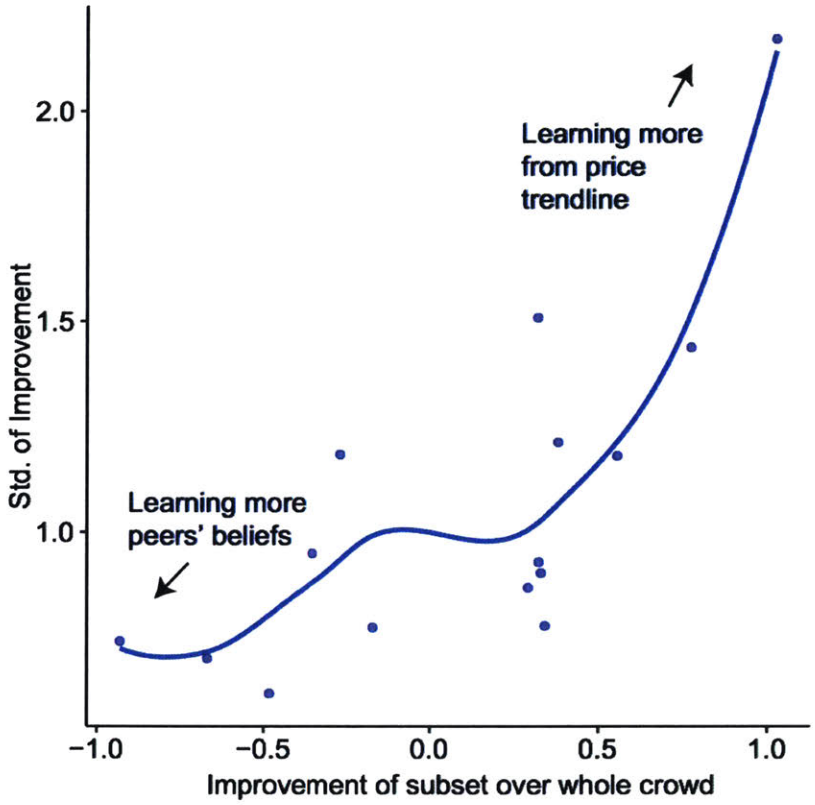


Figure 3-6: In this Pareto curve, we plot the imrovement of various subsets vs. the standard deviation in improvement within this subset. We see a strong risk-return trade-off: predictions made with price history are more accurate, but with higher volatility. The smoothed curved is generated using LOESS[25].

the same subsetting of predictions as described earlier, but only for predictions made during the last week. A plot of the volatility of the asset and futures prices during this week is shown in the appendix.

This last week had only 104 prediction sets (compared to 284 for the first two weeks of data that we previously used for predictions). This last week of data that we use is a disjoint subset of data from the data we previously used. Even though it is a smaller number, it was sufficient to afford us statistically significant results as shown by the 95% confidence intervals of our findings. Again, we bin all α's from the prediction sets during this week (but use a smaller number of bins due to the smaller number of predictions) and investigate the improvements of selected groups compared to the whole crowd.

As can be seen in figure 3-7, we find that our earlier trends are reversed during this period: users who update their predictions when using their peers' beliefs B_H do better (both in terms of the mean improvement and its standard deviation (calculated using bootstraps) than users who use the price history B_T. Conversely, the asset prices were fluctuating wildly during this week which explains why predictions that were modeled using price history are far less accurate and noisy. The improvement values and their accompanying α_s are included in Table 3 in the appendix.

In summary, although it is generally better for users to update their prediction based on the price history, when the price history itself is very uncertain as during instability the week before the Brexit vote, it might be better to pay more attention to peer beliefs.

3.6 Discussion

We show that it is possible to identify updated predictions that are significantly more accurate than the crowd (similar to [70] but they look at resistance to social influence), and do so by estimating which data (social histogram or price history) was used as likelihood to update the prediction. Our result adds a new dimension to understanding the effect of information exposure by showing that exposure to

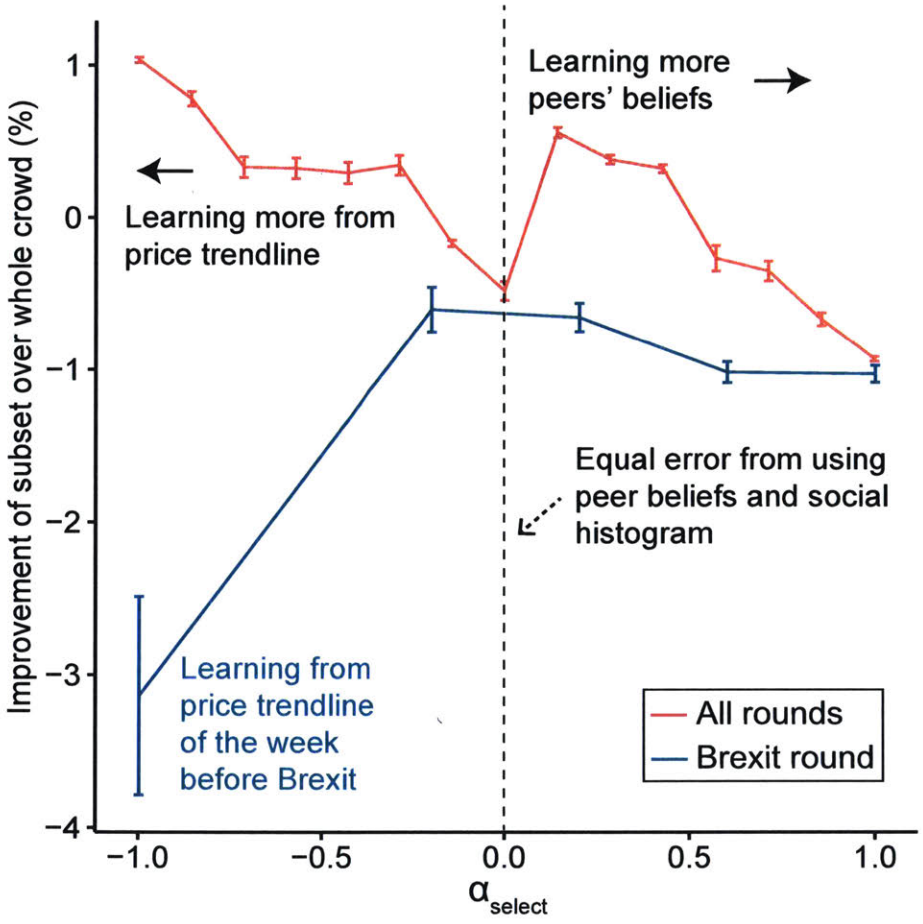


Figure 3-7: We repeat our analysis during the week of data before Brexit when the market was highly volatily. This last week of data that we use is a disjoint subset of data from the data we previously used. We observe that when the price history is very volatile, users who learn from their peers do better than when they learned from the price history. The red line is the same as in Figure 3-5.

ground-truth information (the price history) is also important. Specifically, as shown in our Pareto curve (figure 3-6, we observe that there is a strong risk-reward trade-off between updating beliefs based on social information versus price history.

Beyond shedding light on how exposure to information influences the Wisdom of the Crowd, our main result has implications for the dynamics of financial markets. Speculators trying to predict the future price of assets are expected to use the price history. However, in this work, because we observe that models of the belief update process of individuals using *only* social information (`GaussianSocial`) as their likelihood (evidence) significantly outperform models that use the price history (`GaussianPrice`), this means that individuals strongly prefer learning from their peers instead of from the price data. This is especially surprising as our study's subjects are mid-career financial professionals with years of professional trading experience who would know of the efficient market hypothesis[71, 39] which suggests that the price should incorporate all the market information already and should be the best source of information to update one's belief. Although we observe people to be learning mostly from social information, those who learn from the price history are better at predicting the ground-truth, in agreement with the efficient market hypothesis.

This behavior has been studied widely in financial markets as herding[19], with previous work showing that herding in forecasting is associated with analysts prior forecast errors[24]. Extreme reliance on peer beliefs has been even shown to lead to financial crashes[26]. However, we find that although people are more likely to update their belief using their peers' beliefs, predictions made using the price history (instead of their peers' beliefs) are more accurate, in agreement with the efficient market hypothesis. During periods of external uncertainty (such as during volatile markets), we observe the opposite: listening to one's peers tends to higher accuracy (figure 3-7). Similar behavior has been observed when financial networks are in distress: high clustering and strong tie interaction are observed, with changes in peer-to-peer communication being better at predicting optimality of transactions than prices[99].

Evolutionarily, the fact that individuals prefer learning from the belief of their

peers instead of from the ground truth (price data) at the risk of lower accuracy but with lower variance can be justified by the fact that most people prefer to be conservative in the face of uncertainty. Specifically, negative consequences have a much stronger effect on people's utility[57]. Unfortunately, having a bias towards listening to one's peers (instead of learning from the data) can have quite negative social consequences, for example leading to polarization of opinions[29, 9].

Finally, we observe that the models with strong distributional (`GaussianSocial` assumes all data to be normally distributed and uni-modal) properties are better at modeling the belief update of individuals than the more precise numerical numerical models (`NumericalSocial`). This is in line with the attribute substitution heuristic of human decision-making, whereby people solve a complicated problem by approximating it with a simpler, less accurate model [56]. We even observe that models that assume that the likelihood data is unimodal (when the data is clearly non-unimodal) do better than models that use the smaller, closer or larger mode of the social histogram, another cognitive bias used by people because learning from multi-modal data is cognitively costly[52]. Again, this is surprising as the study's subjects were all mid-career finance professionals, suggesting that even trained quantitive experts performing a familiar task can succumb to the many heuristic and biases common to non-professionals.

3.7 Contribution

In this chapter, we observe that people exhibit strong inductive biases when they learn from social information. We designed a novel study to be able to collect data in order to model people's learning from social information, and we observe that people make strong distributional assumptions when they update their belief based on this information, in addition to preferring to learn from social information.

Chapter 4

Improving Deep Reinforcement Learning at the Individual Level

4.1 Purpose

In this chapter, we present how augmenting a deep reinforcement learning (DRL) model with relational (between agents and objects) inductive biases can significantly improve performance. Such relational descriptions have been observed to be used by humans in learning social relationships.

In recent years, reinforcement learning techniques have enjoyed considerable success in a variety of challenging domains [77], but are typically sample inefficient and often fail to generalize well to even small changes in the environment or task [65]. Humans, by contrast, are able to learn robust skills with orders of magnitude less training. One hypothesis for this discrepancy is that humans view the world in terms of objects and relations between them [13]. Such an inductive bias may be useful reducing sample complexity and improving interpretability and generalization. In this chapter, we show that implementing such an inductive bias not only accelerates and improves learning, but also leads to more interpretable models. This chapter is based on this [3] published paper.

4.2 Background

Deep Learning techniques have proven invaluable in tackling many problems which previously required tricky feature engineering. Reinforcement learning, in particular, has benefited from deep learning, achieving notable successes in a variety of challenging domains [105]. Unfortunately, these approaches are not always sample efficient, often requiring hundreds of thousands of episodes or more of training. More troubling still, current techniques often do not yield general or transferable models [49]. Humans, by contrast, are much better at learning efficient, generalizable strategies from orders of magnitude less data [65]. One potential explanation for this success is that humans view the world in terms of objects and their relations [13]. For example, in viewing an image we may see that the cup is on the table (a *binary relation* between two objects) or the wall is green (a *unary relation* on one object).

This work investigates the effectiveness of the *relational hypothesis* in RL (discussed in more detail in [13]) that a relational inductive bias leads to improvements in learning efficiency, effectiveness, generality and interpretability. Recent work [101, 119, 13, 89] has added support for the relational hypothesis in domains such as visual Q/A [101] and game playing which requires reasoning [119].

In this chapter, we present a novel architecture in which each relation is learned by a separate relational unit which operates on a fixed number of objects (1 or 2 in our experiments) and produces a scalar value in the unit interval[1]. We experiment on a stochastic goal-seeking game in which each episode has a random placement of multiple goal objects, some with positive rewards and some with negative rewards [43]. This game, while simple, is challenging because the agent must learn a location-relative policy to succeed. We find that the relations learned are interpretable. Our model has the ability to learn a binary relation that indicates to what degree an object is, for example, to the right of the agent (it learns relative, not absolute, position relations), while another binary relation can learn 'aboveness' relative to the agent, and a unary relation can be learned that indicates the type of each object.

[1]The logical community refers to these as 'fuzzy' relations.

We show that with a suitable number of relational units (a hyper-parameter of our model), we can achieve performance gains over a multi-headed attention model [119], a standard MLP, a pixel MLP, and the symbolic RL model of [43] on this problem.

4.3 Related Work

Early work in the area of relational reasoning used symbolic representations and tabular learning [34]. More recently [101] introduced a simple relational model with a single relational function for supervised relational learning. A version using the multi-headed attention network [113] is used in [119] to learn a policy and value function. Here, each attention head computes a relation by a applying a scaled dot-product between pairs of objects, normalizing with softmax, and using the result as weighting to update the object representations. Our approach differs in several respects. First, we consider both unary (single argument) and binary (two argument) relations. Second, we allow multiple modules for each type of relation (unary and binary). Third, we do not compute a probability distribution over the attention scores between an object and every other, but instead compute values in the unit interval for each comparison independently. This encodes the intuition that the same relation may hold independently between an object and many others (for example 'above'). Finally, we do not use attention scores to update the object representations but instead supply the concatenated relational values (which we call the *relational state*) directly to the Q-function. Our relational unit can be viewed as implementing a fuzzy logical relation between its arguments. In this respect, our approach connects to the neuro-symbolic reasoning work of [104, 98, 38, 22, 5].

Since we are focusing on the relations between objects in this work, we assume that the objects have already been extracted upstream (for example using the approach of [43] or [64]), and we have available an $m \times 4$ tensor of m object representations ($m-1$ goals, 1 agent). Each object representation is of size 4, consisting of the (x, y) coordinates, the type, and a binary 'existence' marker of whether a goal has been captured by the agent. We consider only agent-object pairs and assume that the

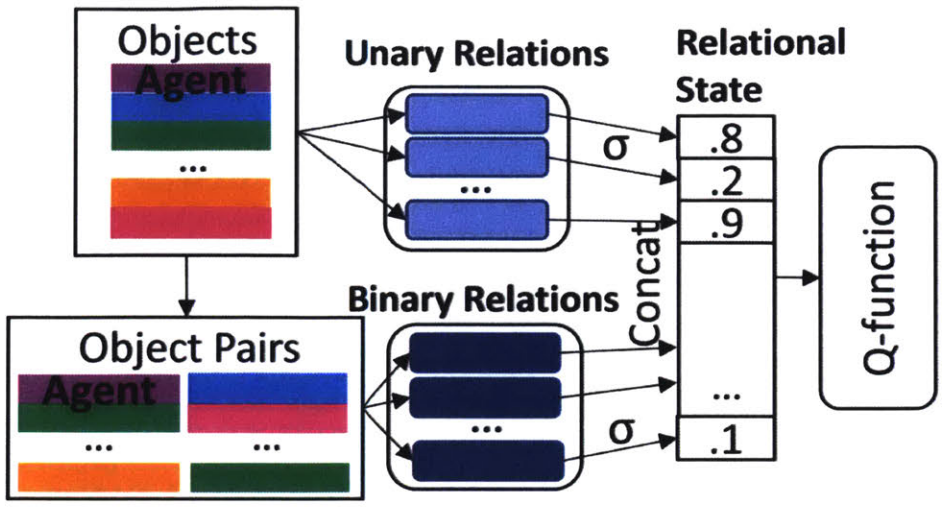


Figure 4-1: The SRN: Object state representations are provided as input and passed directly to each unary unit. Object pairs are computed and passed to each binary unit. The output of the relational units is collected in a *relational state* and supplied as input to a Q-function.

agent representation is supplied first within the tensor of objects.

4.4 Model

As shown in Figure 4-1, our architecture explicitly represents multiple unary (single object) and binary (two objects) relations on objects in the state[2] and concatenates the output values of those relations to form a relational state which is supplied to the Q-function. In essence, our model can be seen as taking the state and pre-processing it into a relational state that can be used with any deep RL algorithm. Specifically, we use differentiable neural relation modules (units which take a vector of objects as input, and output a value in $[0, 1]$) and concatenate their outputs to pass as input to the state action value layers. We call this architecture a *Symbolic Relation Network* (SRN). The unary unit is of the form: $\sigma(W_1 \texttt{ReLU}(W_2 A^T + b_1) + b_2)$ where σ is the sigmoid function, $W_1 \in \mathcal{R}^{1\times k}$ and $W_2 \in \mathcal{R}^{k\times n}$ are weight tensors, $A \in \mathcal{R}^{m\times n}$ is the dimensional input object tensor with m n-dimensional objects, and $b_1 \in \mathcal{R}^{k\times m}$ and

[2]Relations between more than two objects are straightforwardly incorporated into our model but due to the combinatorial increase in computational expense we elected to not considered them in our current work.

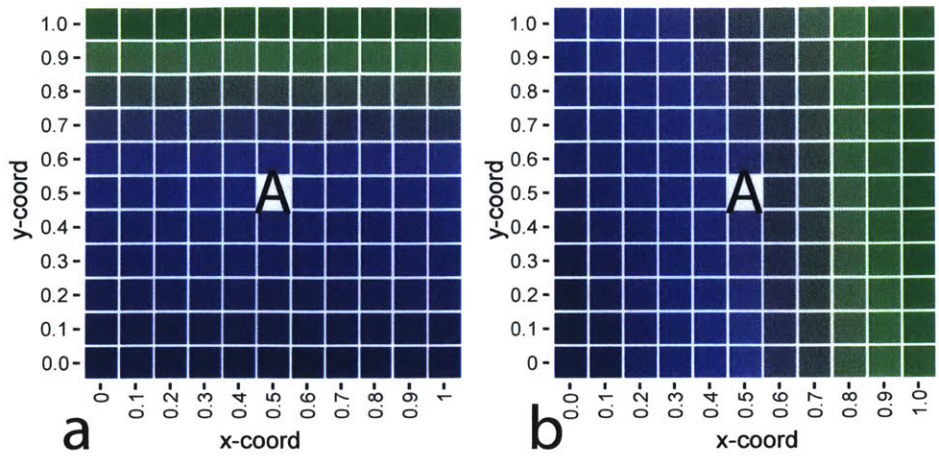


Figure 4-2: Visualization of the output of two binary units when the agent is placed at the center and a goal is placed on each tile which (a) shows the degree of 'aboveness' of the goal from the agent, and (b) shows degree of 'rightness' of the goal from the agent.

$b_2 \in \mathcal{R}^{1\times m}$ are biases. The hidden dimension k is taken to be the same as the object dimension n which is 4 in our experiments. The output is of dimension m, one relation value in $[0, 1]$ for each of the m input objects (i.e. the unit learns a representation of each object). The binary unit is constructed similarly but we form the input tensor by concatenating N pairs of objects to form an $N \times 2n$ input tensor. The state-action value module is a simple linear layer which takes in the concatenation of the relational unit outputs. In implementation, we find that our SRN model can be quite small (< 1000 total parameters) and fast.

4.5 Methods

In this work, we only vary the neural network model that is used to approximate the Q-function $Q(s, a)$, allowing us to fairly compare the different models, while keeping the environment the same[3]. We test the SRN and baseline models using the environment introduced by [43] where an agent can move up, down, left and right and must reach 'good' goals (green circles) that provide a positive reward (+10), and

[3]We plan to share the code for our environment and training procedure, saved trained models, and hyper-parameter configurations.

avoid 'bad' goals (red circles) that provide a negative reward (-10). The game ends when the maximum number of steps is reached (100 in our experiments) or when the agent captures all of the good goals. Each step taken by the agent incurs a small penalty (-0.1). Each episode of training generates a new board with a randomly placed good and bad objects. Our intuition is that a model that learns relationships between each objects, and between objects and the agent, would allow the agent to navigate the board more efficiently to gather the good goals and evade the bad goals.

We use off-the-shelf Deep Q-learning (DQN) [76] as our reinforcement learning algorithm with decaying exploration (parameterized through ϵ-greedy action selection), decaying learning rate for the Adam optimizer [60] and a batch size of 256. We also normalize rewards during learning to be in $[-1, 1]$. We did not find that using the difference of frames and representing goal positions as relative to the agent (all positions fed to our model are absolute) were necessary or helpful for learning.

We employ the testing procedure used in [15, 76, 43]: every 10 training episodes, we pause training and test our model on 10 new random boards for 100 steps (or until the agent finds all the good goals) and record the average test performance over these 10 random boards. We repeat this procedure for at least 10 independent trials and report the average testing performance over these independent trials.

We test our SRN model against a number of baseline models, always ensuring that the baselines have approximately the same number of parameters. Our baselines consists of: 1) a simple Multi-layer Perceptron comprising of three linear layers with ReLU activation learning from the object representation; 2) a relation network based on multi-head dot-product attention [119]; 3) the symbolic RL model of [43]; 4) an MLP which learns directly from pixels instead of from our object based representation consisting of 3 convolutional layers (with ReLU activations and batch normalization in between them) followed by a final linear layer.

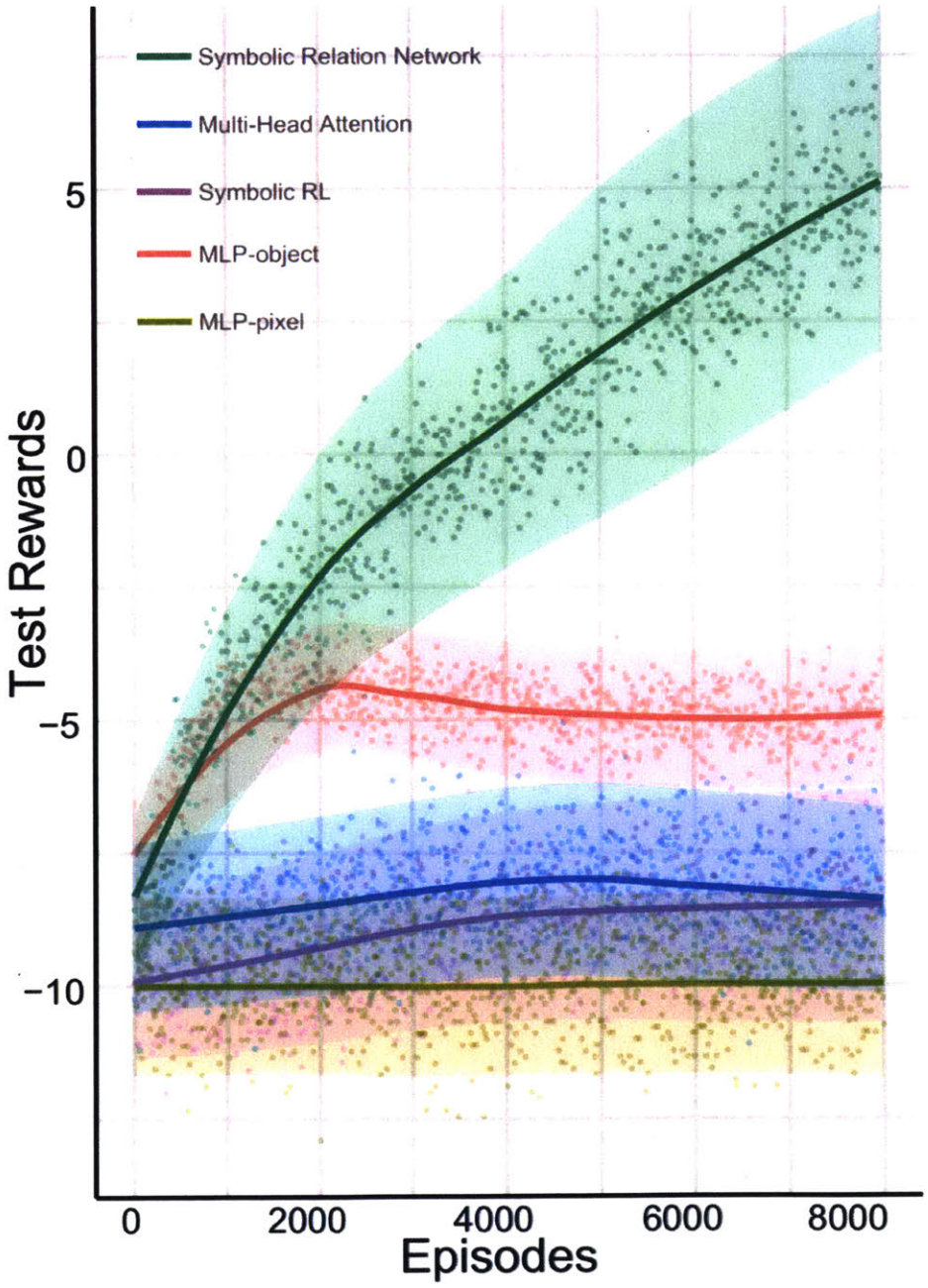


Figure 4-3: Testing performance of different models. Shaded areas show [5-95] confidence interval.

4.6 Results

4.6.1 Interpretability

To evaluate the interpretability of the learned relations, we trained a Symbolic Relation Network with one unary unit and two binary units with DQN on a board with only 1 goal (good and bad) and, after convergence, evaluated the output of each relational unit. We found that the unary unit learns to recognize good and bad objects. Specifically, a unique value between 0 and 1 is learned consistently for each type of goal. We suspect that if goals provided more continuous levels of reward (as opposed to just two levels, good and bad), the unary relationship would learn a more continuous representation of the goal type, but have yet to perform this experiment.

Figure 4-2 shows the evaluation for each binary unit. To create these plots, we placed the agent in the center of the board and evaluated the relationship between the agent and an object placed at each of the colored grid locations. Each such goal location is colored on a blue-green spectrum to indicate the value of the relationship (the output of the binary unit's softmax) when applied to the agent and that goal. Blue represents low values; green represents high values. We evaluated the relation using both good and bad objects and found that the value of the relationship learned was invariant to the type of object. We hypothesize that, since the unary relation learned the type, it was presumably unnecessary for the binary relationship to learn it as well. In the Figure 4-2, (a) shows the agent has learned a notion of degree of goal 'aboveness' of a goal relative to itself while (b) shows the agent has learned a degree of goal 'rightness' relative to itself. Additionally, we found that the same complementary relations can be seen even when the agent is not placed in the center. With these three independent relations, `type`(X), `above`(agent, X), and `right`(agent, X), for objects X, the agent is able to learn a policy that is location relative and hence robust to the random placement of the objects and agent each game.

Model	Reward	Recall	Precision	F1
SRN	**7.27**	**0.50**	0.71	**0.59**
MHA	-5.07	0.31	**0.75**	0.44
Symbolic RL	-5.51	0.32	0.69	0.44
MLP-objects	-3.04	0.35	0.52	0.41
MLP-pixel	-7.35	0.17	0.29	0.21

Table 4.1: Performance of the Symbolic Relation Network compared to other baselines.

4.6.2 Learning Performance

As can be seen in Figure 4-3, our Symbolic Relation Network (with 8 binary units and 1 unary unit) obtains the highest reward (statistically significant over a [5-95] confidence interval) and outperforms all of the baseline models of the same size (same number of parameters), namely a multi-layer perceptron, a multi-head dot-product attention (MHA) [119] and the symbolic RL model of [43]. In some exploratory experiments, we observe that MHA models with 100 times the number of parameters as our SRN can start competing in performance with our SRN. We also compare our SRN against a larger (with 100 times the number of parameters) Multi-layer Perceptron that learns directly from pixels, and find very little learning (in agreement with [43]).

We also compute three other metrics for performance: the *recall* (proportion of good goals found out of the total number of good goals), the *precision* (proportion of good goals found out of the total number of goals found) and their harmonic mean (F1 score), $F1 = \left(\frac{recall^{-1}+precision^{-1}}{2}\right)^{-1}$ which is a standard combined measure. As we can see in Table 4.1, our model outperforms other baseline model in terms of reward, recall, and F1 score and is close to MHA in precision. The results show that our Symbolic Relation Network not only strongly outperforms other models of comparable size in both reward and F1 measures, but is also more interpretable, learning three complementary relations useful for solving the task.

4.7 Contribution

In this chapter we introduced a neural architecture for relation learning with multiple unary and binary relational units. Our experiments on a simple but challenging stochastic, goal-seeking environment show that the right relational inductive bias leads to higher performance over a multi-headed attention model, a standard MLP, a pixel MLP and a symbolic RL model.

Chapter 5

Improving Deep Reinforcement Learning at the Collective Level

5.1 Purpose

Now that we know that there are cognitive limits on social information and how humans adapt their learning, can we apply some of these insights to modern deep reinforcement learning algorithms? This is especially important because every distributed algorithm relies on an implicit communication network between the processing units being used in the algorithm, and there is a lot of potential to wire and organize these learning agents less naively.

Given these cognitive limits, humans and animal species have evolved a number of adaptations in order to optimally search a parameter landscape for the best performing set. One of these adaptations is the use of certain sparse topologies[11] of communication as a way to trade-off exploration and exploitation of parameters. This chapter is based on this [2] published paper.

5.2 Background

Every distributed algorithm relies on an implicit communication network between the processing units being used in the algorithm. In the case of distributed machine

learning, these units pass information such as data, parameters, or rewards between each other. For example, in the popular A3C [75] reinforcement learning algorithm, multiple 'workers' are spawned with local copies of a global neural network, and they are used to collectively update the global network. These workers can be either viewed as simply implementing the parallelized form of an algorithm, or they can be seen as a type of multi-agent distributed optimization approach to searching the reward landscape for parameters that maximize performance.

In this work, we take the latter approach of thinking of the 'workers' as separate agents that can search a reward landscape more or less efficiently. We adopt such an approach because it allows us to consider improvements studied in the field of multi-agent optimization [41], specifically the literatures of networked optimization (optimization over networks of agents with local rewards) [81, 82, 80] and collective intelligence (the study of mechanisms of how agents learn, influence and collaborate with each other) [117, 118]. These two literatures suggest a number of different ways to improve such multi-agent optimization, and, in this work, we choose to focus on one of main ways to do so: optimizing the topology of communication between agents (i.e. the local and global characterization of which neighbors each agent can share data, parameters, or rewards with).

We focus on communication topology because it has been shown to result in increased exploration, higher overall maximum reward, and higher diversity of solutions in both simulated high-dimensional optimization problems [66] and human experiments [11], and because, to the best of our knowledge, almost no prior work has investigated how the topology of communication between agents affects learning performance in distributed Deep Reinforcement Learning (DRL). The two topologies that are almost always used are either a complete (fully-connected) network, in which all processors communicate with each other; or a star network—in which all processors communicate with a single hub server.

Here, we empirically investigate whether using alternative communication topologies between agents could lead to improving learning performance in the context of DRL. Given that network effects are sometimes only significant with large numbers

of agents, we choose to build upon one of the DRL algorithms most oriented towards parallelizability and scalability: Evolution Strategies [96, 102, 115, 100]. We introduce Networked Evolution Strategies (NetES), a networked decentralized variant of ES. NetES, like many DRL algorithms and evolutionary methods, relies on aggregating the rewards from a population of processors that search in parameter space to optimize a single global parameter set. Using NetES, we explore how the communication topology of a population of processors affects learning performance.

Key aspects of our approach, findings, and contributions are as follows:

- We introduce the notion of communication network topologies to the ES paradigm for DRL tasks.

- We perform an ablation study using various baseline controls to make sure that any improvements we see come from using alternative topologies and not other factors.

- We compare the learning performance of the main topological families of communication graphs, and observe that one family (Erdos-Renyi graphs) does best.

- Using an optimized Erdos-Renyi graph, we evaluate NetES on five difficult DRL benchmarks and find large improvements compared to using a fully-connected communication topology. We observe that our 1000-agent Erdos-Renyi graph can compete with 3000 fully-connected agents.

- We derive an upper bound which provides theoretical insights into why alternative topologies might outperform a fully-connected communication topology. We find that our upper bound only depends on the topology of learning agents, and not on the reward function of the reinforcement learning task at hand, which indicates that our results likely will generalize to other learning tasks.

5.3 Preliminaries

5.3.1 Evolution Strategies for Deep RL

As discussed earlier, given that network effects are sometimes only significant with large numbers of agents, we choose to build upon one of the DRL algorithms most oriented towards parallelizability and scalability: Evolution Strategies.

We begin with a brief overview of the application of the Evolution Strategies (ES) [102] approach to deep reinforcement learning, following Salimans et al. [100]. Evolution Strategies is a class of techniques to solve optimization problems by utilizing a derivative-free parameter update approach. The algorithm proceeds by selecting a fixed model, initialized with a set of weights $\boldsymbol{\theta}$ (whose distribution p_ϕ is parameterized by parameters ϕ), and an objective (reward) function $R(\cdot)$ defined externally by the DRL task being solved. The ES algorithm then maximizes the average objective value $\mathbb{E}_{\boldsymbol{\theta}\sim p_\phi} R(\boldsymbol{\theta})$, which is optimized with stochastic gradient ascent. The score function estimator for $\nabla_\phi \mathbb{E}_{\boldsymbol{\theta}\sim p_\phi} R(\boldsymbol{\theta})$ is similar to REINFORCE [116], given by $\nabla_\phi \mathbb{E}_{\boldsymbol{\theta}\sim p_\phi} R(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}\sim p_\phi}[R(\boldsymbol{\theta})\nabla_\phi \log p_\phi(\boldsymbol{\theta})]$.

As introduced by [100], the update equation used in this algorithm for the parameter $\boldsymbol{\theta}$ at any iteration $t+1$, for an appropriately chosen learning rate α and noise standard deviation σ, is a discrete approximation to the gradient:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \frac{\alpha}{N\sigma^2} \sum_{i=1}^{N} \left(R(\boldsymbol{\theta}^{(t)} + \sigma \boldsymbol{\epsilon}_i^{(t)}) \cdot \sigma \boldsymbol{\epsilon}_i^{(t)} \right) \tag{5.1}$$

This update rule is implemented by spawning a collection of N agents at every iteration t, with perturbed versions of $\boldsymbol{\theta}^{(t)}$, i.e. $\{(\boldsymbol{\theta}^{(t)} + \sigma\boldsymbol{\epsilon}_1^{(t)}), ..., (\boldsymbol{\theta}^{(t)} + \sigma\boldsymbol{\epsilon}_N^{(t)})\}$ where $\epsilon \sim \mathcal{N}(0, I)$. The algorithm then calculates $\boldsymbol{\theta}^{(t+1)}$ which is broadcast again to all agents, and the process is repeated.

In summary, either a centralized controller or each agent holds a global parameter θ, records the perturbed noise $\boldsymbol{\epsilon}_i^{(t)}$ used by *all* agents, collects rewards from *all* agents at the end of an episode, calculates the approximated gradient and obtains a new global parameter θ. Because each agent uses information from *all* other agents to

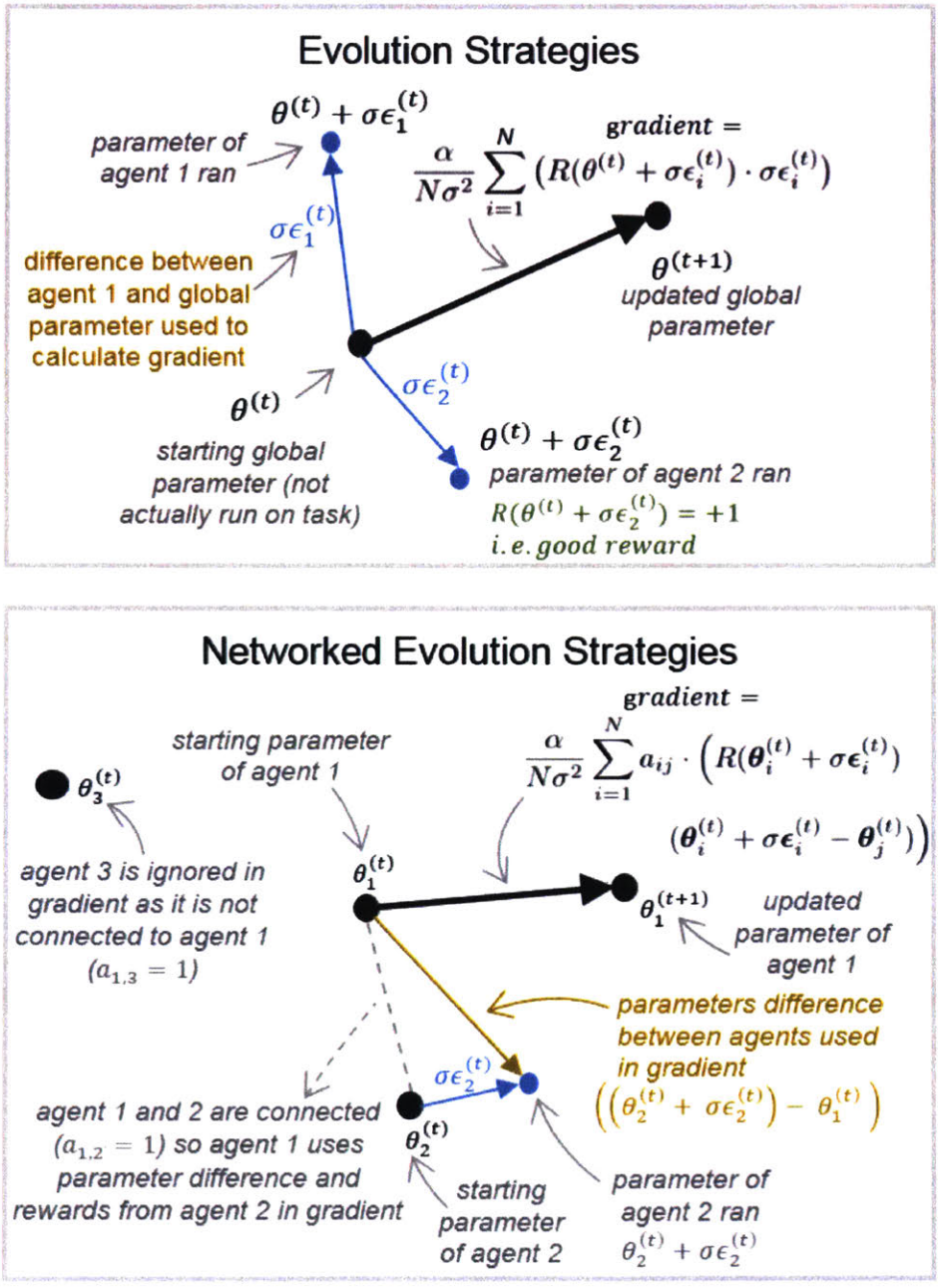


Figure 5-1: **All**: Each black dot is a parameter set held by an agent and each blue dot is a perturbed parameter set being run on the DRL task. **Top**: Evolution strategies where the gradient is the average over the difference in parameters (if all agents have the same parameters, this difference is just the noise) weighted by rewards. **Bottom**: Networked Evolution Strategies where the gradient for agent 1 is still the average over difference in parameters but only over agents that are connected (not agent 3 in this case).

update their parameter, the algorithm uses a fully-connected (complete) network. And because they all use the same information, they come to consensus to the same global parameter each round and therefore only a single $\theta^{(t)}$ parameter is needed to be expressed in the algorithm. Each agent therefore only deals with one-step perturbations of the global parameter, $(\theta^{(t)} + \sigma\epsilon_i^{(t)})$.

Through equation 5.1, each agent is taking a weighted average of the differences (perturbations) between their last local parameter copy and the perturbed copies of each agent, (the differences being $\sigma\epsilon_i^{(t)} = ((\theta^{(t)} + \sigma\epsilon_i^{(t)}) - \theta^{(t)})$) where the weight is given by the reward at the location of each perturbed copy $R(\boldsymbol{\theta}^{(t)} + \sigma\boldsymbol{\epsilon}_i^{(t)})$.

However, when agents are not arranged in a fully-connected network topology, even if all the agents start with the same global parameter $\boldsymbol{\theta}^{(t_0)}$, after the very first update step, they would each hold different parameters $\boldsymbol{\theta}_j^{(t_0+1)}$ as each agent's gradient would be calculated using a unique subset of its neighbors rewards and parameters. This is illustrated in Fig 5-1. In developing NetES, we will therefore have to make explicit the local versions of the parameter $\boldsymbol{\theta}_j^{(t)}$. When each agent has a local copy of the parameter, $\boldsymbol{\theta}_i^{(t)}$, the weighted average (using the same weights $R(\boldsymbol{\theta}^{(t)} + \sigma\boldsymbol{\epsilon}_i^{(t)})$) is still, as in the standard case, over the differences between their last local parameter and the perturbed copies of each agent. Because each agent now has different parameters, this difference is $((\theta_i^{(t)} + \sigma\epsilon_i^{(t)}) - \theta_j^{(t)})$. In this notation, Equation 5.1 is then:

$$\boldsymbol{\theta}_j^{(t+1)} = \boldsymbol{\theta}_j^{(t)} + \frac{\alpha}{N\sigma^2}\sum_{i=1}^{N}\left(R(\boldsymbol{\theta}_i^{(t)} + \sigma\boldsymbol{\epsilon}_i^{(t)}) \cdot (\boldsymbol{\theta}_i^{(t)} + \sigma\boldsymbol{\epsilon}_i^{(t)} - \boldsymbol{\theta}_j^{(t)})\right) \tag{5.2}$$

5.4 Problem Statement

The task ahead is to take the standard ES algorithm and operate it over new communication topologies, wherein each agent is only allowed to communicate with its neighbors. This would allows us to then see if any topologies perform better than the de-facto fully-connected topology. The ultimate goal would be to optimize over

Type	Task	Fully-connected	Erdos	Improv. %
MuJoCo	Ant-v1	4496	4938	**9.8**
MuJoCo	HalfCheetah-v1	1571	7014	**346.3**
MuJoCo	Hopper-v1	1506	3811	**153.1**
MuJoCo	Humanoid-v1	762	6847	**798.6**
Roboschool	Humanoid-v1	364	429	**17.9**

Table 5.1: Improvements from Erdos-Renyi networks with 1000 nodes compared to fully-connected networks.

the space of all possible topologies to find the ones that perform best for our task at hand - an interesting possibility for future work, but outside the scope of our work. Instead, we take as a more tractable starting point a comparison of four popular graph families (including the fully-connected topology).

5.4.1 NetES : Networked Evolution Strategies

We denote a network topology by $\boldsymbol{A} = \{a_{ij}\}$, where $a_{ij} = 1$ if agents i and j communicate with each other, and equals 0 otherwise. $\boldsymbol{A}$ represents the *adjacency matrix* of connectivity, and fully characterizes the communication topology between agents. In a fully connected network, we have $a_{ij} = 1$ for all i, j.

Using adjacency matrix $\boldsymbol{A}$, it is straightforward to allow equation 5.2 to operate over any communication topologies:

$$\boldsymbol{\theta}_j^{(t+1)} = \boldsymbol{\theta}_j^{(t)} + \frac{\alpha}{N\sigma^2} \sum_{i=1}^{N} a_{ij} \cdot \left(R(\boldsymbol{\theta}_i^{(t)} + \sigma \boldsymbol{\epsilon}_i^{(t)}) \cdot (\boldsymbol{\theta}_i^{(t)} + \sigma \boldsymbol{\epsilon}_i^{(t)} - \boldsymbol{\theta}_j^{(t)}) \right) \tag{5.3}$$

Because equation 5.3 uses the same weighted average as in ES (equations 5.1 and 5.2), when fully-connected networks are used (i.e. $a_{ij} = 1$) and when agents start with the same parameters, equation 5.3 reduces to 5.1.

The only other change (other than using a_{ij} in the update rule) introduced by NetES is the use of periodic global broadcasts. We implemented parameter broadcast as follows: at every iteration, with a probability p_b (in practice we set it to 0.8, a popular hyperparameter value in other algorithms), we choose to replace all

agents' current parameters with the best agent's performing weights, and then continue training (as per Equation 5.3) after that. The same broadcast techniques have been used in many other algorithms to balance local vs. global search (e.g. 'exploit' in Population-based Training [55] by replacing current weights with weights that give the highest rewards).

Given the three additions of NetES to ES (the use of alternate topologies through a_{ij}, the use of different parameters, and broadcast), we run careful controls during an ablation study to investigate where the improvement in learning we observe come from - we show later that they come from the use of alternative topologies as shown in see Fig. 5-2B.

5.4.2 Communication topologies under consideration

Given the update rule as per equation 5.3, the goal is then to find which topology leads to the highest improvement. Because we are drawing inspiration from the study of collective intelligence and networked optimization, we use topologies that are prevalent in modeling how humans and animals learn collectively:

- **Erdos-Renyi Networks:** Networks where each edge between any two nodes has a fixed independent probability of being present [36], which are among the commonly used benchmark graphs for comparison in social networks [84].

- **Scale-Free Networks:** Scale-free networks, whose degree distribution follows a power law [23], are commonly observed in citation and signaling biological networks[10].

- **Small-World Networks:** Networks where most nodes can be reached through a small number of neighbors, resulting in the famous 'six degrees of separation' [109].

- **Fully-Connected Networks:** Networks where every node is connected to every other node.

Each of these network families can be parameterized by the number of nodes N, and their degree distribution, and we can randomly sample instances of graphs from each family. Erdos-Renyi networks, for example, are parameterized by their average density p ranging from 0 to 1, where 0 would lead to a completely disconnected graph (no nodes are connected), and 1 would lead back to a fully-connected graph. The lower p is, the sparser a randomly generated network is. Similarly, the degree distribution of scale-free networks is defined by the exponent of the power distribution. Because each graph is generated randomly, two graphs with the same parameters will be different if they have different random seeds, even though, on average, they will have the same average degree (and therefore the same number of links).

5.4.3 Consequences of update rule

Previous work [11] demonstrates that the exact form of the update rule does not matter much because sparser networks are better as long as the distributed strategy is to find and aggregate the parameters with the highest reward (as opposed to, for example, finding the most common parameters many agents hold). Therefore, although our update rule is a straightforward extension of ES, we expect that our primary insight—that network topology can affect deep reinforcement learning—to still be useful with alternative update rules.

Secondly, although Equation 5.3 is a biased gradient estimate, at least in the short term, it is unclear whether in practice we achieve a biased or an unbiased gradient estimate, marginalizing over time steps between broadcasts. This is because in the full algorithm (algorithm 1) we implement, we combine this update rule with a periodic parameter broadcast (as is common in distributed learning algorithms - we will address this in detail in a later section), and every broadcast returns the agents to a consensus position.

Future work can better characterize the theoretical properties of NetES and similar networked DRL algorithms using the recently developed tools of calculus on networks (e.g., [1]). Empirically, we find that NetES achieves large performance improvements.

5.4.4 Predicted improved performance of NetES

Through the modifications to ES we have described, we are now able to operate on any communication topology. Due to previous work in networked optimization and collective intelligence which shows that alternative network structures result in better performance, we expect NetES to perform better on DRL tasks when using alternative topologies compared to the de facto fully-connected topology. We also expect to see differences in performance between families of topologies.

5.5 Related Work

There have been many variants of Evolution Strategies over the years, such as CMA-ES [6] which also updates the covariance matrix of the Gaussian distribution, Natural Evolution strategies [115] where the inverse of the Fisher Information Matrix of search distributions is used in the gradient update rule, and, of course, the Evolution Strategies of Salimans et al. [100] (which we build on) which was modified for scalability in DRL. However, in all the approaches described above, agents are organized in an implicit fully-connected centralized topology.

A focus of recent DRL has been the ability to be able to run more and more agents in parallel (i.e. scalability). An early example is the Gorila framework [79] that collects experiences in parallel from many agents. Another is A3C [75] that we discussed earlier. IMPALA [37] is a recent algorithm which solves many tasks with a single parameter set. Population Based Training [55] optimizes both learning weights and hyperparameters. Again, these algorithms implicitly use a fully-connected topology between learning agents.

There has also been work in the multi-agent reinforcement learning literature focusing on how independent agents can solve competitive and collaborative problems. For example, recent work investigated the role communication topology, but it is focused on agents solving different tasks [120]. One recent study [69] investigated the effect of communication network topology, but only as an aside, and on very small networks - and they also observe improvements when using not fully-connected

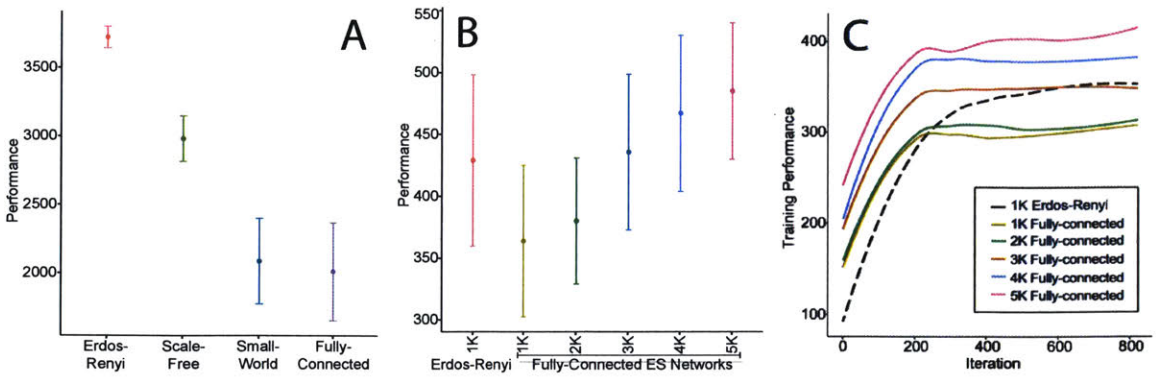


Figure 5-2: **A**: Learning performance on all network families: Erdos-Renyi graphs do best, fully-connected graphs do worst (MuJoCo Ant-v1 task with small networks of 100 nodes). **B**: Evaluation results for Erdos-Renyi graph with 1000 agents compared to fully-connected networks with varying network sizes (RoboSchool Humanoid-v1). **C**: Comparing Erdos-Renyi graph with 1000 agents to fully-connected networks with varying network sizes on training (not evaluation metric) performance (Roboschool Humanoid-v1). **All**: Error bars represent 95% confidence intervals.

networks.

On the other hand, work in the networked optimization literature has demonstrated that the network structure of communication between nodes significantly affects the convergence rate and accuracy of multi-agent learning [81, 82, 80]. However this work has been focused on solving global objective functions that are the sum (or average) of private, local node-based objective functions - which is not always an appropriate framework for deep reinforcement learning. In the collective intelligence literature, alternative network structures have been shown to result in increased exploration, higher overall maximum reward, and higher diversity of solutions in both simulated high-dimensional optimization [66] and human experiments [11].

To the best of our knowledge, no prior work has focused on investigating how the topology of communication between agents affects learning performance in distributed DRL, for large networks and on popular graph families.

5.6 Experimental Procedure

5.6.1 Goal of experiments

The main goal of this work is to run ES on DRL tasks but using alternative topologies through our networked variant of ES, NetES, and to see if alternative topologies (instead of the de-facto fully-connected topology) perform better. Therefore, we want to be able to generate communication topologies from each of the four popular random graph families, wire our agents using this topology and deploy them to solve the DRL task at hand.

Algorithm 1: Networked Evolution Strategies

Input: Learning rate α, noise standard deviation σ, initial policy parameters $\boldsymbol{\theta}_i^{(0)}$ where $i = 1, 2, \ldots, \text{N}$ (for N workers), adjacency matrix $\boldsymbol{A}$, global broadcast probability p_b
Initialize: n workers with known random seeds, initial parameters $\boldsymbol{\theta}_i^{(0)}$
for $t = 0, 1, 2, \ldots$ **do**
  for each worker $i = 1, 2, \ldots, \text{N}$ **do**
    Sample $\boldsymbol{\epsilon}_j^{(t)} \sim \mathcal{N}(0, I)$
    Compute returns $R_i = R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)})$
  Sample $\beta^{(t)} \sim \mathcal{U}(0, 1)$
  if $\beta^{(t)} < p_b$ **then**
    Set $\boldsymbol{\theta}_i^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}_i^{(t)}} R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)})$
  else
    for each worker $i = 1, 2, \ldots, \text{n}$ **do**
      Set $\boldsymbol{\theta}_i^{(t+1)} \leftarrow \boldsymbol{\theta}_i^{(t)} + \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \left(R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) \cdot (\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \right)$

5.6.2 Procedure

We evaluate our NetES algorithm on a series of popular benchmark tasks for deep reinforcement learning, selected from two frameworks—the open source Roboschool [88] benchmark, and the MuJoCo framework [107]. The five benchmark tasks we evaluate on are: Humanoid-v1 (Roboschool and Mujoco), HalfCheetah-v1 (MuJoCo), Hopper-v1 (MuJoCo) and Ant-v1 (MuJoCo). Our choice of benchmark tasks is motivated by the difficulty of these walker-based problems.

To maximize reproducibility of our empirical results, we use the standard evaluation metric of collecting the total reward agents obtain during a test-only episode, which we compute periodically during training [76, 15, 100]. Specifically, with a probability of 0.08, we intermittently pause training, take the parameters of the best agent and run this parameter (without added noise perturbation) for 1000 episodes, and take the average total reward over all episodes—as in Salimans et al. [100]. When performance eventually stabilizes to a maximum 'flat' line (determined by calculating whether a 50-episode moving average has not changed by more than 5%), we record the maximum of the evaluation performance values for this particular experimental run. As is usual [15], training performance (shown in Fig. 5-2C) will be slightly lower that the corresponding maximum evaluation performance (shown in Table 5.1). We observe this standard procedure to be quite robust to noise.

We repeat this evaluation procedure for multiple random instances of the same network topology by varying the random seed of network generation. These different instances share the same average density p (i.e. the same average number of links) and the same number of nodes N. Since each node runs the same number of episode time steps per iteration, different networks with the same p can be fairly compared. For all experiments (all network families and sizes of networks), we use an average network density of 0.2 because it is sparse enough to provide good learning performance, and consistent (not noisy) empirical results.

We then report the average performance over 6 runs with 95% confidence intervals. We share the JSON files that fully describe our experiments and our anonymized code at `www.bit.ly/2Dsk2OJ`.

In addition to using the evaluation procedure of Salimans et al. [100], we also use their exact same neural network architecture: multilayer perceptrons with two 64-unit hidden layers separated by *tanh* nonlinearities. We also keep all the modifications to the update rule introduced by Salimans et al. to improve performance: (1) training for one complete episode for each iteration; (2) employing antithetic or mirrored sampling, also known as mirrored sampling [44], where we explore $\epsilon_i^{(t)}, -\epsilon_i^{(t)}$ for every sample $\epsilon_i^{(t)} \sim \mathcal{N}(0, I)$; (3) employing fitness shaping [115] by applying a

rank transformation to the returns before computing each parameter update, and (4) weight decay in the parameters for regularization. We also use the exact same hyper-parameters as the original OpenAI (fully-connected and centralized) implementation [100], varying only the network topology for our experiments.

5.7 Results

5.7.1 Empirical performance of network families

We first use one benchmark task (MuJoCo Ant-v1, because it runs fastest) and networks of 100 agents to evaluate NetES on each of the 4 families of communication topology: Erdos-Renyi, scale-free, small-world and the standard fully-connected network. As seen in Fig 5-2A, two topologies outperform fully-connected networks: Erdos-Renyi and Scale-Free networks. We also establish that, on this task, Erdos-Renyi strongly outperforms the other topologies and we decide to focus on Erdos-Renyi graphs for all other results going forward - this choice is supported by our theoretical results which indicate that Erdos-Renyi would do better on any task.

5.7.2 Empirical performance on all benchmarks

Using Erdos-Renyi networks (as they previously performed best compare to other network families), we run larger networks of 1000 agents on all 5 benchmark results. As can be seen in Table 5.1, our Erdos-Renyi networks outperform fully-connected networks on all benchmark tasks, resulting in improvements ranging from 9.8% on MuJoCo Ant-v1 to 798% on MuJoCo Humanoid-v1. All results are statistically significant (based on 95% confidence intervals).

We note that the difference in performance between Erdos-Renyi and fully-connected networks is higher for smaller networks (Fig. 5-2A and Fig. 5-3B) compared to larger networks (Table 5.1) for the same benchmark, and we observe this behavior across different benchmarks. We believe that this is because NetES is able to achieve higher performance with fewer agents due to its efficiency of exploration, as supported in

our empirical and theoretical results below.

5.7.3 Varying network sizes

So far, we have compared alternative network topologies with fully-connected networks containing the same number of agents. In this section, we investigate whether organizing the communication topology using Erdos-Renyi networks can outperform larger fully-connected networks. We choose one of the benchmarks that had a small difference between the two algorithms at 1000 agents, Roboschool Humanoid-v1. As shown in Fig. 5-2B and the training curves (which display the training performance, not the evaluation metric results which would be higher as discussed earlier) in Fig. 5-2C, an Erdos-Renyi network with 1000 agents provides comparable performance to 3000 agents arranged in a fully-connected network.

5.7.4 Ablation Study

To ensure that none of the modifications we implemented in the ES algorithm are causing improvements in performance, instead of just the use of alternative network topologies, we run control experiments on each modification: 1) the use of broadcast, 2) the fact that each agent/node has a different parameter set. We test all combinations.

Broadcast effect

We want to make sure that broadcast (over different probabilities ranging from 0.0 to 1.0) does not explain away our performance improvements. We compare 'disconnected' networks, where agents can only learn from their own parameter update and from broadcasting (they do not see the rewards and parameters of any other agents each step as in NetES). We compare them to Erdos-Renyi networks and fully-connected networks of 1000 agents on the Roboschool Humanoid-v1 task. As can be seen in Fig. 5-3A practically no learning happens with **just** broadcast and no network. These experiments show that broadcast does not explain away the performance

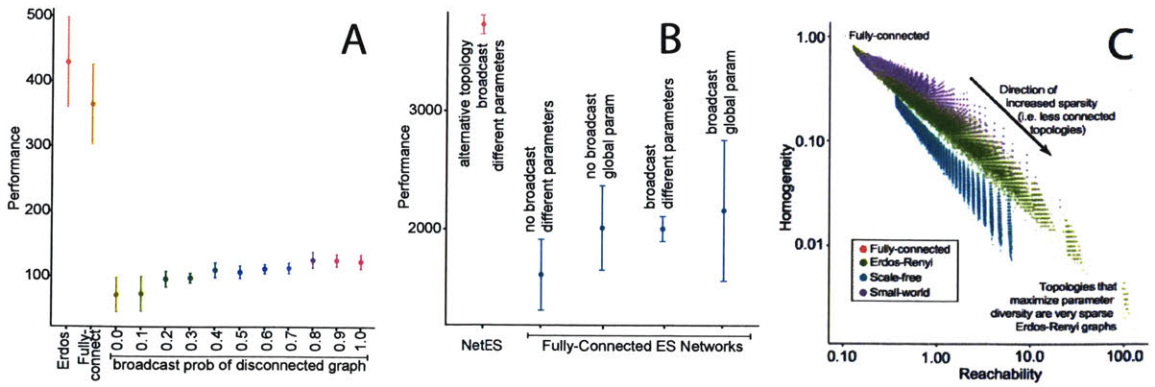


Figure 5-3: **A**: Agents with any amount of periodic broadcasting do not learn (RoboSchool Humanoid-v1 with 1000 agents). **B**: None of the control baselines with fully-connected networks learn, showing that the use of alternative topologies is what leads to learning (MuJoCo Ant-v1 with 100 agents). **C**: We generate instances of random networks from our four families of networks, and observe that sparser Erdos-Renyi graphs maximize the diversity of parameter updates.

improvement we observe when using NetES.

Global versus individual parameters

The other change we introduce in NetES is to have each agent hold their own parameter value $\boldsymbol{\theta}_i^{(t)}$ instead of a global (noised) parameter $\boldsymbol{\theta}^{(t)}$. We therefore investigate the performance of the following 4 control baselines: fully-connected ES with 100 agent running: (1) same global parameter, no broadcast; (2) same global parameter, with broadcast; (3) different parameters, with broadcast; (4) different parameters, no broadcast; compared to NetES running an Erdos-Renyi network. For this experiment we use MuJoCo Ant-v1. As shown in Fig 5-3B, NetES does better than all 4 other control baselines, showing that the improvements of NetES come from using alternative topologies and not from having different local parameters for each agent.

5.8 Theoretical Insights

In this section, we present theoretical insights into why alternative topologies can outperform fully-connected topologies, and why Erdos-Renyi networks also outperform the other two network families we have tested. A motivating factor for introducing

alternative connectivity and having each agent hold local parameters (as per Equation 5.3) is to search the parameter space more completely, a common motivation in DRL and optimization in general. One possible heuristic for measuring the capacity to explore the parameter space is the diversity of parameter updates during each iteration, which can be measured by the variance of parameter updates:

Theorem 1. *In a NetES update iteration t for a system with N agents with parameters $\Theta = \{\theta_1^{(t)}, ..., \theta_N^{(t)}\}$, agent communication matrix $\boldsymbol{A} = \{a_{ij}\}$, agent-wise perturbations $\mathcal{E} = \{\epsilon_1^{(t)}, ..., \epsilon_N^{(t)}\}$, and parameter update $u_i^{(t)} = \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \big(R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) \cdot ((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) - (\boldsymbol{\theta}_i^{(t)}))\big)$ as per Equation 5.3, the following relation holds:*

$$\mathsf{Var}_i[u_i^{(t)}] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \Big\{ \Big(\frac{\|A^2\|_F}{(\min_l |\boldsymbol{A}_l|)^2} \Big) \cdot f(\Theta, \mathcal{E}) - \Big(\frac{\min_l |\boldsymbol{A}_l|}{\max_l |\boldsymbol{A}_l|} \Big)^2 \cdot \frac{\sigma^2}{N} \Big(\sum_{i,j} \boldsymbol{\epsilon}_i^{(t)} \boldsymbol{\epsilon}_j^{(t)} \Big) \Big\} \tag{5.4}$$

Here, $|\boldsymbol{A}_l| = \sum_j a_{jl}$, and $f(\Theta, \mathcal{E}) = \sqrt{\big(\sum_{j,k,m} \big((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_m^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_m^{(t)})\big)^2\big)}$.

The proof for Theorem 2 is provided in the appendix.

This theoretical upper-bound is merely expository; it is not indicative of the *worst-case* performance, which requires the optimization of a lower-bound. We use this theoretical insight to understand the *capacity* for parameter exploration supplied by any network topology and not to **choose** the best network topology (which would require a lower bound). It is also important to note that the quantity in Theorem 2 is not the **variance of the value function gradient**, which is typically minimized in reinforcement learning. It is instead the **variance in the positions in parameter space** of the agents after a step of our algorithm. This quantity is more productively conceptualized as akin to a radius of exploration for a distributed search procedure rather than in its relationship to the variance of the gradient. The challenge is then to maximize the search radius of positions in parameter space to find high-performing parameters. As far as the side effects this might have, given the common wisdom that increasing the variance of the value gradient in single-agent reinforcement learning

can slow convergence, it is worth noting that noise (i.e. variance) is often critical for escaping local minima in other algorithms, e.g. via stochasticity in SGD.

By Theorem 2, we see that the diversity of exploration in the parameter updates across agents is likely affected by two quantities that involve the connectivity matrix A: the first being the term $(\|A^2\|_F/(\min_l |\boldsymbol{A}_l|))^2$ (henceforth referred to as the *reachability* of the network), which according to our bound we want to maximize, and the second being $(\min_l |\boldsymbol{A}_l|/\max_l |\boldsymbol{A}_l|)^2$ (henceforth referred to as the *homogeneity* of the network), which according to our bound we want to be as small as possible in order to maximize the diversity of parameter updates across agents. Reachability and homogeneity are not independent, and are statistics of the degree distribution of a graph. It is interesting to note that the upper bound *does not depend on the reward landscape* $R(\cdot)$ *of the task at hand*, indicating that our theoretical insights should be independent of the learning task.

Reachability is the squared ratio of the total number of paths of length 2 in A to the minimum number of links of all nodes of A. The sparser a network, the larger the reachability. For Erdos-Renyi graphs, $(\|A^2\|_F/(\min_l |\boldsymbol{A}_l|))^2 \approx (pN)^{-1/2}$, where p is the average density of the network (the inverse of sparsity), the probability that any two nodes being connected. Homogeneity is the squared ratio of the minimum to maximum connectivity of all nodes of A: the higher this value, the more homogeneously connected the graph is. The sparser a network is, the lower is the homogeneity of a network. In the case of Erdos-Renyi networks, $(\min_l |\boldsymbol{A}_l|/\max_l |\boldsymbol{A}_l|)^2 \approx 1 - 8\sqrt{(1-p)/(Np)}$ (the proofs and plots for Erdos-Renyi are provided in the supplementary material).

Using the above definitions for reachability and homogeneity, we generate random instances of each network family, and plot them in Fig. 5-3C. Two main observations can be made from this result: (1) Erdos-Renyi networks maximize reachability and minimize homogeneity, which means that they likely maximize the diversity of parameter exploration. (2) Fully-connected networks are the single worst network in terms of exploration diversity (they minimize reachability and maximize homogeneity, the opposite of what would be required for maximizing parameter exploration according

to the suggestion of our bound). These theoretical results agree with our empirical results: Erdos-Renyi networks perform best, followed by scale-free networks, while fully-connected networks perform worst.

5.9 Contribution

In chapter 2, we showed that an inductive bias in reinforcement learning can improve learning performance. In chapter 3, we observed the existence of cognitive limits on social learning, which can be alleviated by certain inductive biases, discussed in chapter 3. An under-explored aspect of inductive bias in machine learning is the use of inter-agent inductive biases: perhaps learning via some specific network topology, or some specific way to communicate parameters, rewards or actions would be most helpful?

In this chapter, we showed that using human-inspired network topologies can have a huge improvement on learning performance. We did so by extending ES, a DRL algorithm, to use alternative network topologies and empirically showed that the conventional fully-connected topology performs worse in our experiments. We also provided an theoretical insights into why alternative topologies may be superior.

Chapter 6

Conclusion and Future Work

6.1 Contributions

Below is a summary of the contributions of this thesis:

- Cognitive Limits on Social Learning:
  - In this chapter, we observe that even financial traders – whose performance depends on their ability to learn new strategies from from others – exhibit cognitive constraints on the number of social connections that they have. Interestingly, their performance is not correlated with their congnitive limit, but is however strongly predicted by their amount of exploration.
  - We also build bots that transcend human congnitive limits, we show that they can outperform humans even when using very simple bot trading strategies.
- Modeling Social Learning:
  - Given that people exhibit strong cognitive constraints in how they learn from social information, the question is: how do such constraints mold the way people learn socially? Through a large-scale novel study, we model

people's belief update process after being exposed to dynamic social information and price history using Bayesian models of cognition.

  - We observe many inductive biases, such as the fact that people make strong distributional assumptions on the distribution of the belief of their peers, and that they prefer to learn from social information than from non-social data.
  - We observe the trade-offs of learning from social data as a risk-reward tradeoff.

- Social Inductive Bias for DRL at the Individual Level:
  - In this chapter, we present how augmenting a deep reinforcement learning model with relational (between agents and objects) inductive biases can significantly improve performance. Such relational descriptions have been observed to be used by humans in learning social relationships.
  - Our experiments on a simple but challenging stochastic, goal-seeking environment show that the right relational inductive bias leads to higher performance over a multi-headed attention model, a standard MLP, a pixel MLP and a symbolic RL model.
  - We also find that our inductive bias leads to an interpretable model of relationships between objects.

- Social Inductive Bias for DRL at the Collective Level:
  - We introduce the notion of communication network topologies to the ES paradigm for DRL tasks.
  - We perform an ablation study using various baseline controls to make sure that any improvements we see come from using alternative topologies and not other factors.
  - We compare the learning performance of the main topological families of communication graphs, and observe that one family (Erdos-Renyi graphs) does best.

- Using an optimized Erdos-Renyi graph, we evaluate NetES on five difficult DRL benchmarks and find large improvements compared to using a fully-connected communication topology. We observe that our 1000-agent Erdos-Renyi graph can compete with 3000 fully-connected agents.
- We derive an upper bound which provides theoretical insights into why alternative topologies might outperform a fully-connected communication topology. We find that our upper bound only depends on the topology of learning agents, and not on the reward function of the reinforcement learning task at hand, which indicates that our results likely will generalize to other learning tasks.

6.2 Future Work

This thesis hopes to have provided some insights into the inductive biases humans and bots use for social learning. Given a future increasingly filled with decentralized autonomous machine learning systems, there is an increasing need to understand social learning to build resilient, scalable and effective learning systems. Some of the most exciting avenues of research to be explored are discussed below.

6.2.1 Consciousness

Humans are not just problem solving machine. Most of our experience of existence is not problem solving, but subjective experience. There is active work in trying to understand how do tasks transition from unconscious to conscious in neuroscience through predictive coding[95] or conscious ignition[32]. We know very little about how consciousness works from a computational and statistical perspective. We know that 'access consciousness', which is what individuals report they have access to consciously can hold 5-9 items (e.g. names, objects, and other simple concepts) or about 40 bits/sec of information[108]. There is vast amount of active work into finding signatures of consciousness from brain signals[12].

However, it remains to be seen how to replicate some of this ability into artificial learning systems. What advatages would consciousness give us for increased reasoning ability? Computationally, there is some preliminary work investigating these ideas in machine learning[16], and it is fertile ground for innovation.

6.2.2 Theory of Mind

So far, we have focused on modeling the process of how agents actively learn from each other. A related and profound question is: how do people model the minds of other people? This is an active and very promising area of research for social learning.

One active area of research is modeling Theory of Mind as a Bayesian process[8]. Another is to train agents to model under agent's mind using neural networks for making fast inference[94]. A novel area of research is to investigate the recursive theory of mind problem[61].

6.2.3 Causality

Causality can be regarded as one of the most foundational inductive biases, and it is surprisingly absent in machine learning. Allowing machine agents to reason causally would allow for a wealth of useful functionality:

- The ability to reason counter-factually: what would happen if a different action was taken, without having to take this action? This is a growing area of research in reinforcement learning where it is important to understand how to explore and experiment in the world efficiently and for maximum reward or minimum regret[21, 30].

- Machine learning is mostly concerned with prediction and inference, but not with understanding interventional causality[92] whose goal is to understand the precise causal mechanism that underpins the data generating distribution observed. There is interesting new work trying to take advantage of causality for faster transfer learning[18]

- Humans have created a collective, organized and compressed systematic enterprise to causally understand the world: science. We could use machine learning in many ways to enrich and extend the current scientific enterprise, or perhaps, even more ambitiously, to endow artificial intelligence with the goals of creating science.

There is much to be done. I hope that this thesis has pushed enough against the walls and made, at least, a dent.

Appendix 1: Inductive Biases in Social Learning

Performance over all rounds

Here we report the values of the residual for each round for all models. We can observe that `GaussianSocial` does best.

MODEL	ROUND						
	1 (S&P 500)	2 WTI Oil	3 Gold	7 (S&P 500)	8 (S&P 500)	9 (S&P 500)	12 (S&P 500)
GaussianSocial	1.53 (0.19)	3.97 (0.48)	1.08 (0.13)	0.92 (0.04)	0.70 (0.04)	1.51 (0.07)	1.23 (0.13)
GaussianSocialModes	1.94 (0.20)	4.85 (0.54)	1.30 (0.19)	1.24 (0.05)	0.98 (0.04)	1.88 (0.08)	1.64 (0.13)
NumericalSocial	2.01 (0.23)	5.24 (0.61)	1.60 (0.25)	1.52 (0.08)	1.07 (0.06)	2.31 (0.10)	2.31 (0.22)
NumericalPrice	2.25 (0.23)	8.70 (0.87)	2.64 (0.19)	1.57 (0.08)	1.09 (0.06)	2.36 (0.10)	2.75 (0.23)
GaussianPrice	2.46 (0.24)	10.3 (0.92)	2.70 (0.22)	1.59 (0.07)	1.13 (0.06)	2.41 (0.10)	2.72 (0.22)
DeGroot	2.04 (0.22)	5.32 (0.60)	1.52 (0.13)	1.71 (0.07)	1.17 (0.06)	2.51 (0.09)	2.27 (0.21)

Table 1: Values of the residual for each round for all models. Numbers in parentheses show the 95% error.

Brexit

The week before the Brexit vote caused high volatility in the price of the S&P 500 as shown in Figure -1. The Brexit vote happened on the last day of the round. 104 new predictions were made the last week of the round.

Table of Improvements by subsetting

In this section, we report the improvement $(\overline{B^{S_{\alpha s}}_{post}} - \overline{B^{S_{all}}_{post}})/\overline{B^{S_{all}}_{post}}$ when selecting a subset of users.

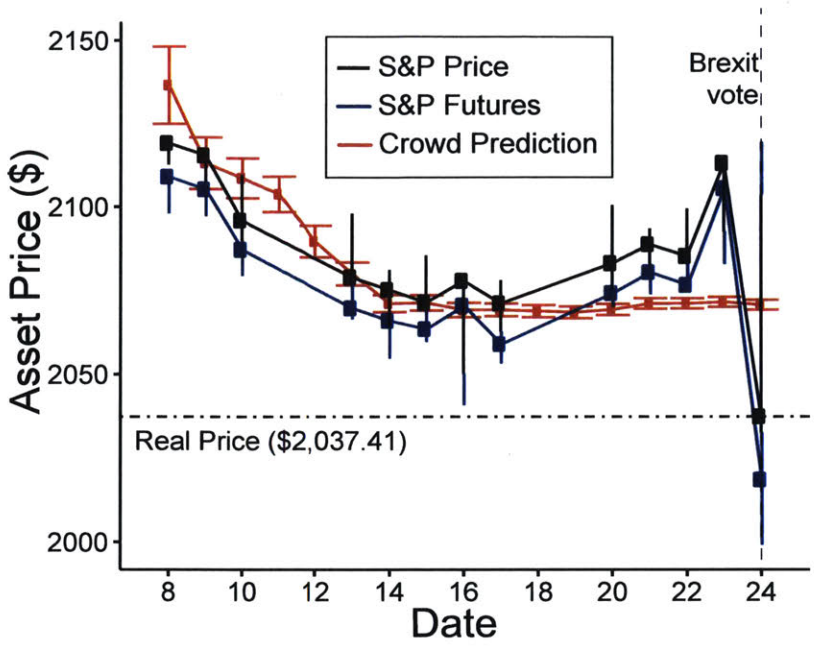


Figure -1: The mean cumulative prediction of the users during the brexit round is shown in red, with error bars for the 95% confidence interval. The close, low and high price of the asset and its underlying futures is shown in the other two candlestick plots. The asset and futures overestimate the price and then crash during the last week.

α_s	**Improvement (%)** $(\overline{B_{post}^{S_{\alpha_s}}} - \overline{B_{post}^{S_{all}}})/\overline{B_{post}^{S_{all}}}$	**95% CI**
-1.0	1.03	0.02
-0.9	0.77	0.05
-0.7	0.33	0.07
-0.6	0.32	0.07
-0.4	0.29	0.07
-0.3	0.34	0.06
-0.1	-0.17	0.02
0.0	-0.48	0.06
0.1	0.56	0.03
0.3	0.38	0.03
0.4	0.32	0.03
0.6	-0.27	0.08
0.7	-0.35	0.06
0.9	-0.67	0.04
1.0	-0.93	0.02

Table 2: Improvements achieve by subsetting predictions via α_s for all rounds. Confidence interval are calculated through 100 bootstraps.

α_s	**Improvement (%)** $(\overline{B^{S_{\alpha_s}}_{post}} - \overline{B^{S_{all}}_{post}})/\overline{B^{S_{all}}_{post}}$	**95% CI**
-1.0	-3.14	0.65
-0.2	-0.61	0.15
0.2	-0.66	0.09
0.6	-1.02	0.07
1.0	-1.03	0.05

Table 3: Improvements achieve by subsetting predictions via α_s only for predictions the week before Brexit. Confidence interval are calculated through 100 bootstraps.

Appendix 2: Proof for Theorem 2

Here we provide proofs Theorem 1 from the main paper concerning the diversity of the parameter updates.

Theorem 2. *In a multi-agent evolution strategies update iteration t for a system with N agents with parameters* $\Theta = \{\theta_1^{(t)}, ..., \theta_N^{(t)}\}$, *agent communication matrix* $\boldsymbol{A} = \{a_{ij}\}$, *agent-wise perturbations* $\mathcal{E} = \{\epsilon_1^{(t)}, ..., \epsilon_N^{(t)}\}$, *and parameter update* $u_i^{(t)}$ *given by the sparsely-connected update rule:*

$$u_i^{(t)} = \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \left(R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) \cdot ((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) - (\boldsymbol{\theta}_i^{(t)}))\right)$$

The following relation holds:

$$\mathsf{Var}_i[u_i^{(t)}] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \Big\{ \Big(\frac{\|A^2\|_F}{(\min_l |\boldsymbol{A}_l|)^2} \Big) \cdot f(\Theta, \mathcal{E}) - \Big(\frac{\min_l |\boldsymbol{A}_l|}{\max_l |\boldsymbol{A}_l|} \Big)^2 \cdot g(\mathcal{E}) \Big\} \quad (1)$$

Here, $|\boldsymbol{A}_l| = \sum_j a_{jl}$, $f(\Theta, \mathcal{E}) = \Big(\sum_{j,k,m}^{N,N,N} \big((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_m^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_m^{(t)}) \big)^2 \Big)^{\frac{1}{2}}$, *and* $g(\mathcal{E}) = \frac{\sigma^2}{N} \Big(\sum_{i,j}^{N,N} \boldsymbol{\epsilon}_i^{(t)} \boldsymbol{\epsilon}_j^{(t)} \Big)$.

Proof. From Equation 2, the update rule is given by:

$$u_i^{(t)} = \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \left(R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) \cdot ((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) - (\boldsymbol{\theta}_i^{(t)}))\right) \quad (2)$$

The variance of $u_i^{(t)}$ can be written as:

$$\mathsf{Var}_i[u_i^{(t)}] = \mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})^2] - (\mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})])^2 \quad (3)$$

Expanding $\mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})^2]$:

$$= \frac{1}{N} \sum_{i \in \mathcal{A}} \Big\{ \frac{\gamma}{N\sigma^2} \sum_{j=1} \boldsymbol{a}_{ij} \cdot R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) \cdot (\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \Big\}^2 \quad (4)$$

Simplifying:

$$= \frac{1}{N\sigma^4} \sum_{i,j,k} \Big(\frac{\boldsymbol{a}_{ij}\boldsymbol{a}_{ik}}{|\boldsymbol{A}_i|^2} R(\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)}) R(\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)}) \cdot (\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_i^{(t)}) \Big) \quad (5)$$

Since $R(\cdot) \leq \max R(\cdot)$, therefore:

$$\leq \frac{\max^2 R(\cdot)}{N\sigma^4} \sum_{i,j,k} \frac{\boldsymbol{a}_{ij}\boldsymbol{a}_{ik}}{|\boldsymbol{A}_i|^2} \cdot (\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_i^{(t)}) \quad (6)$$

$$\leq \frac{\max^2 R(\cdot)}{N\sigma^4} \sum_{i,j,k} \frac{\boldsymbol{a}_{ij}\boldsymbol{a}_{ik}}{\min_l |\boldsymbol{A}_l|^2} \cdot (\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_i^{(t)}) \quad (7)$$

By the Cauchy-Schwarz Inequality:

$$\mathbb{E}_{i\in\mathcal{A}}[(u_i^{(t)})^2] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \Big(\sum_{i,j,k} \frac{(\boldsymbol{a}_{ij}\boldsymbol{a}_{ik})^2}{\min_l |\boldsymbol{A}_l|^4} \Big)^{\frac{1}{2}} \cdot \Big(\sum_{i,j,k} \big((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_i^{(t)}) \big)^2 \Big)^{\frac{1}{2}} \quad (8)$$

Since $\boldsymbol{a}_{ij} \in \{0,1\} \forall\ (i,j), (\boldsymbol{a}_{ij}\boldsymbol{a}_{ik})^2 = \boldsymbol{a}_{ij}\boldsymbol{a}_{ik}\ \forall (i,j,k)$. Additionally, we know that $\boldsymbol{a}_{ij} = \boldsymbol{a}_{ji}$, since $\boldsymbol{A}$ is symmetric. Therefore, $\sum_i \boldsymbol{a}_{ij}\boldsymbol{a}_{ik} = \sum_i \boldsymbol{a}_{ji}\boldsymbol{a}_{ik} = \boldsymbol{A}^2_{jk}$. Using this:

$$\mathbb{E}_{i\in\mathcal{A}}[(u_i^{(t)})^2] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \cdot \Big(\frac{|\boldsymbol{A}^2|^{\frac{1}{2}}}{\min_l |\boldsymbol{A}_l|^2} \Big) \cdot \Big(\sum_{i,j,k} \big((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_i^{(t)}) \big)^2 \Big)^{\frac{1}{2}} \quad (9)$$

Replacing $\Big(\sum_{i,j,k} \big((\boldsymbol{\theta}_j^{(t)} + \sigma\boldsymbol{\epsilon}_j^{(t)} - \boldsymbol{\theta}_i^{(t)}) \cdot (\boldsymbol{\theta}_k^{(t)} + \sigma\boldsymbol{\epsilon}_k^{(t)} - \boldsymbol{\theta}_i^{(t)}) \big)^2 \Big)^{\frac{1}{2}} = f(\Theta, \mathcal{E})$, where $\Theta = \{\boldsymbol{\theta}_i^{(t)}\}_{i=1}^N, \mathcal{E} = \{\boldsymbol{\epsilon}_i\}_{i=1}^N$ for compactness, we obtain:

$$\mathbb{E}_{i\in\mathcal{A}}[(u_i^{(t)})^2] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \cdot \Big(\frac{|\boldsymbol{A}^2|^{\frac{1}{2}}}{\min_l |\boldsymbol{A}_l|^2} \Big) \cdot f(\Theta, \mathcal{E}) \quad (10)$$

Similarly, the squared expectation of $(u_i^{(t)})$ over all agents can be given by:

$$(\mathbb{E}_{i\in\mathcal{A}}[u_i^{(t)}])^2 = \Big(\frac{1}{N}\sum_{i\in\mathcal{A}}\{\frac{\gamma}{N\sigma^2}\sum_{j=1}\boldsymbol{a}_{ij}\cdot R(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)}) \cdot(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)}-\boldsymbol{\theta}_i^{(t)})\}\Big)^2 \tag{11}$$

$$= \frac{1}{N^2\sigma^4}\Big(\sum_{i\in\mathcal{A}}\{\frac{1}{|\boldsymbol{A}_i|}\sum_{j=1}\boldsymbol{a}_{ij}\cdot R(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)})\cdot(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)}-\boldsymbol{\theta}_i^{(t)})\}\Big)^2 \tag{12}$$

$$= \frac{1}{N^2\sigma^4}\Big(\sum_{i,j}\{\frac{\boldsymbol{a}_{ij}}{|\boldsymbol{A}_i|}\cdot R(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)})\cdot(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)}-\boldsymbol{\theta}_i^{(t)})\}\Big)^2 \tag{13}$$

Since $R(\cdot)\geq\min R(\cdot)$, therefore:

$$\geq \frac{\min^2 R(\cdot)}{N^2\sigma^4}\Big(\sum_{i,j}\{\frac{\boldsymbol{a}_{ij}}{|\boldsymbol{A}_i|}\cdot(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)}-\boldsymbol{\theta}_i^{(t)})\}\Big)^2 \tag{14}$$

$$\geq \frac{\min^2 R(\cdot)}{N^2\sigma^4\max_l|\boldsymbol{A}_l|^2}\Big(\sum_{i,j}\{\boldsymbol{a}_{ij}\cdot(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)}-\boldsymbol{\theta}_i^{(t)})\}\Big)^2 \tag{15}$$

Since $\boldsymbol{A}$ is symmetric, $\sum_{i,j}^{N,N}\boldsymbol{a}_{ij}\cdot(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j-\boldsymbol{\theta}_i^{(t)}) = \sum_{i,j}^{N,N}\boldsymbol{a}_{ij}\cdot(\boldsymbol{\theta}_i^{(t)}+\sigma\boldsymbol{\epsilon}_i-\boldsymbol{\theta}_j^{(t)})$. Therefore:

$$= \frac{\min^2 R(\cdot)}{N^2\sigma^4\max_l|\boldsymbol{A}_l|^2}\Big(\sum_{i,j}\frac{1}{2}\{\boldsymbol{a}_{ij}\cdot(\boldsymbol{\theta}_j^{(t)}+\sigma\boldsymbol{\epsilon}_j^{(t)}-\boldsymbol{\theta}_i^{(t)}) + \boldsymbol{a}_{ij}\cdot(\boldsymbol{\theta}_i^{(t)}+\sigma\boldsymbol{\epsilon}_i^{(t)}-\boldsymbol{\theta}_j^{(t)})\}\Big)^2 \tag{16}$$

Therefore,

$$(\mathbb{E}_{i\in\mathcal{A}}[u_i^{(t)}])^2 = \frac{\min^2 R(\cdot)}{N^2\sigma^2 \max_l |\boldsymbol{A}_l|^2}\Big(\sum_{i,j}\frac{1}{2}\{\boldsymbol{a}_{ij}\cdot(\boldsymbol{\epsilon}_j^{(t)}+\boldsymbol{\epsilon}_i^{(t)})\}\Big)^2 \quad (17)$$

Using the symmetry of $\boldsymbol{A}$, we have that $\sum_{i,j}^{N,N}\boldsymbol{a}_{ij}\boldsymbol{\epsilon}_i = \sum_{i,j}^{N,N}\boldsymbol{a}_{ij}\boldsymbol{\epsilon}_j$. Therefore:

$$= \frac{\min^2 R(\cdot)}{N^2\sigma^2 \max_l |\boldsymbol{A}_l|^2}\Big(\sum_{i,j}\boldsymbol{a}_{ij}\cdot\boldsymbol{\epsilon}_j^{(t)}\Big)^2 \quad (18)$$

$$= \frac{\min^2 R(\cdot)}{N^2\sigma^2 \max_l |\boldsymbol{A}_l|^2}\Big(\sum_{j}|\boldsymbol{A}_j|\cdot\boldsymbol{\epsilon}_j^{(t)}\Big)^2 \quad (19)$$

$$\geq \frac{\min^2 R(\cdot)\min_l |\boldsymbol{A}_l|^2}{N^2\sigma^2 \max_l |\boldsymbol{A}_l|^2}\Big(\sum_{i,j}\boldsymbol{\epsilon}_i^{(t)}\boldsymbol{\epsilon}_j^{(t)}\Big) \quad (20)$$

Combining both terms of the variance expression, and using the normalization of the iteration rewards that ensures $\min R(\cdot) = -\max R(\cdot)$, we can obtain (using $g(\mathcal{E}) = \frac{\sigma^2}{N}\Big(\sum_{i,j}\boldsymbol{\epsilon}_i^{(t)}\boldsymbol{\epsilon}_j^{(t)}\Big)$):

$$\mathsf{Var}_{i\in\mathcal{A}}[u_i^{(t)}] \leq \frac{\max^2 R(\cdot)}{N\sigma^4}\Big\{\Big(\frac{|\boldsymbol{A}^2|^{\frac{1}{2}}}{\min_l |\boldsymbol{A}_l|^2}\Big)\cdot f(\Theta,\mathcal{E}) - \Big(\frac{\min_l |\boldsymbol{A}_l|^2}{\max_l |\boldsymbol{A}_l|^2}\Big)\cdot g(\mathcal{E})\Big\} \quad (21)$$

□

Appendix 2 : Approximating Reachability and Homogeneity for Large Erdos-Renyi Graphs

Recall that a Erdos-Renyi graph is constructed in the following way

1. Take n nodes
2. For each pair of nodes, link them with probability p

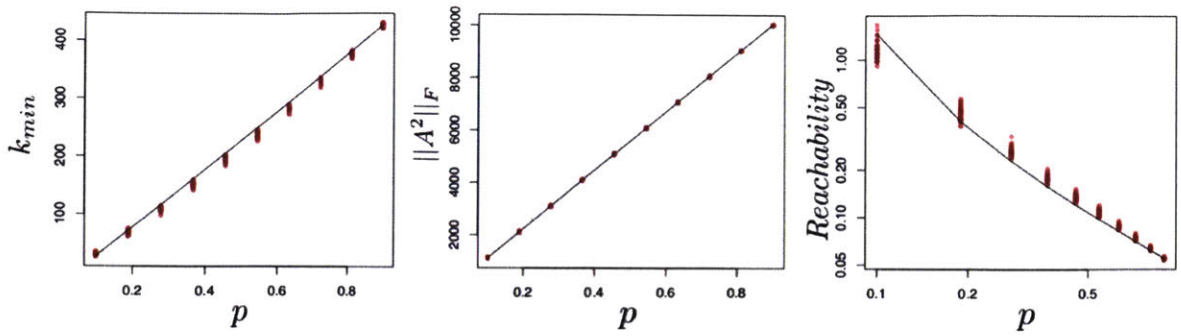


Figure -2: Comparison between the values of k_{min}, $||A^2||_F$, and Reachability as a function of p for different realizations of the Erdos-Renyi model (points) and their approximations given in Equations (23), (22) and (24) respectively (lines).

The model is simple, and we can infer the following:

- The average degree of a node is $p(n-1)$
- The distribution of degree for the nodes is the Binomial distribution of $n-1$ events with probability p, $B(n-1,p)$.
- The (average) number of paths of length 2 from one node i to a node $j \neq i$ ($n_{ij}^{(2)}$) can be calculated this way: a path of length two between i and j involves a third node k. Since there are $n-2$ of them, the maximun number of paths between i and j is $n-2$. However, for that path to exists there has to be a link between i and k and k and j, an event with probability p^2. Thus, the average number of paths between i and j is $p^2(n-2)$

Estimating Reachability

We can then estimate Reachability:

$$Reachability = \frac{||A^2||_F}{(min_l|A_l|)^2} = \frac{\sqrt{\sum_{i,j} n_{ij}^{(2)}}}{k_{min}^2}$$

where $k_{min} = (min_l|A_l|)$ is the minimum degree in the network. Given the above calculations we can approximate

$$\sum_{i,j} n_{ij}^{(2)} = \sum_{i} n_{ii}^{(2)} + \sum_{i \neq j} n_{ij}^{(2)} \approx n \times [p(n-1)] + n(n-1) \times [p^2(n-2)]$$

where the first term is the number of paths of length 2 from i to i summed over all nodes, i.e. the sum of the degrees in the network. The second term is the sum of $p^2(n-2)$ for the terms in which $i \neq j$. For large n we have that

$$\sum_{i,j} n_{ij}^{(2)} \approx p^2 n^3$$

and thus,

$$||A^2||_F \approx \sqrt{p^2 n^3}. \tag{22}$$

For the denominator k_{min} we could use the distribution of the minimum of the binomial distribution $B(n-1,p)$. However, since it is a complicated calculation we can approximate this way: since the binomial distribution $B(n-1,p)$ looks like a Gaussian, we can say that the minimum of the distribution is closed to the mean minus two times the standard deviation:

$$k_{min} \approx p(n-1) - 2\sqrt{p(n-1)(1-p)} \tag{23}$$

Once again in the case of large n we have

$$k_{min} \approx pn$$

Thus

$$Reachability \approx \frac{\sqrt{p^2 n^3}}{[p(n-1) - 2\sqrt{p(n-1)(1-p)}]^2} \tag{24}$$

As we can see in the figure those approximations work very well for realizations of the Erdos-Renyi networks.

Assuming that n is large, we can approximate

$$Reachability \approx \frac{pn^{3/2}}{p^2n^2} = \frac{1}{pn^{1/2}}$$

Thus the bound decreases with increasing n and p. Note that the density of the Erdos-Renyi graph (the number of links over the number of possible links) is p. And thus for a fixed n more sparse networks $p \simeq 0$ have larger Reachability than more connected networks $p \simeq 1$.

Estimating Homogeneity

The Homogeneity is defined as

$$Homogeneity = \left(\frac{k_{min}}{k_{max}}\right)^2$$

As before we can approximate

$$k_{max} \approx p(n-1) + 2\sqrt{p(n-1)(1-p)}$$

And thus

$$Homogeneity \approx \left(\frac{p(n-1) - 2\sqrt{p(n-1)(1-p)}}{p(n-1) + 2\sqrt{p(n-1)(1-p)}}\right)^2$$

For large p we can approximate it to be

$$Homogeneity \approx 1 - 8\frac{\sqrt{1-p}}{\sqrt{np}} \tag{25}$$

which shows that for $p \simeq 1$ we have that Homogeneity grows as a function of p. Thus for fixed number of nodes n, increasing p we get larger values of the Homogeneity. See figure 2

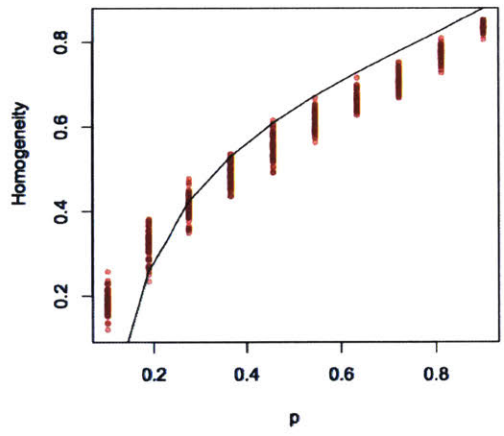


Figure -3: Comparison for the Homogeneity in the Erdos-Renyi case for different values of p and $n = 500$. Points correspond to the real data, while the lines are the approximations given by Equation (25).

Bibliography

[1] Daron Acemoglu, Munther A Dahleh, Ilan Lobel, and Asuman Ozdaglar. Bayesian learning in social networks. *The Review of Economic Studies*, 78(4):1201–1236, 2011.

[2] Dhaval Adjodah, Dan Calacci, Abhimanyu Dubey, Peter Krafft, Esteban Moro, and AlexSandy' Pentland. How to organize your deep reinforcement learning agents: The importance of communication topology. 2018.

[3] Klinger Tim Adjodah, Dhaval and Josh Joseph. Symbolic relation networks for reinforcement learning. 2018.

[4] Kenneth J Arrow, Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O Ledyard, Saul Levmore, Robert Litan, Paul Milgrom, Forrest D Nelson, et al. The promise of prediction markets, 2008.

[5] Masataro Asai and Alex Fukunaga. Classical planning in deep latent space: Bridging the subsymbolic-symbolic boundary. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[6] Anne Auger and Nikolaus Hansen. A restart cma evolution strategy with increasing population size. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, pages 1769–1776. IEEE, 2005.

[7] Bahador Bahrami. optimally interacting minds. 329(September):1081–1086, 2011.

[8] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[9] Delia Baldassarri and Peter Bearman. Dynamics of political polarization. *American sociological review*, 72(5):784–811, 2007.

[10] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[11] Daniel Barkoczi and Mirta Galesic. Social learning strategies modify the effect of network structure on group performance. *Nature communications*, 7, 2016.

[12] Pablo Barttfeld, Lynn Uhrig, Jacobo D Sitt, Mariano Sigman, Béchir Jarraya, and Stanislas Dehaene. Signature of consciousness in the dynamics of resting-state brain activity. *Proceedings of the National Academy of Sciences*, 112(3):887–892, 2015.

[13] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018.

[14] Joshua Becker, Devon Brackbill, and Damon Centola. Network dynamics of social influence in the wisdom of crowds. *Proceedings of the national academy of sciences*, 114(26):E5070–E5076, 2017.

[15] 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.

[16] Yoshua Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.

[17] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[18] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

[19] Sushil Bikhchandani and Sunil Sharma. Herd behavior in financial markets. *IMF Staff papers*, 47(3):279–310, 2000.

[20] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[21] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *arXiv preprint arXiv:1811.06272*, 2018.

[22] Andres Campero, Aldo Pareja, Tim Klinger, Josh Tenenbaum, and Sebastian Riedel. Logical rule induction and theory learning using neural theorem proving, 2018.

[23] Krzysztof Choromański, Michał Matuszak, and Jacek Miekisz. Scale-free graph with preferential attachment and evolving internal vertex structure. *Journal of Statistical Physics*, 151(6):1175–1183, 2013.

[24] Michael B Clement and Senyo Y Tse. Financial analyst characteristics and herding behavior in forecasting. *The Journal of finance*, 60(1):307–341, 2005.

[25] William S Cleveland and Susan J Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610, 1988.

[26] Giancarlo Corsetti, Paolo Pesenti, and Nouriel Roubini. What caused the asian currency and financial crisis? *Japan and the world economy*, 11(3):305–373, 1999.

[27] Iain D Couzin. Collective cognition in animal groups. *Trends in cognitive sciences*, 13(1):36–43, 2009.

[28] Fiery Cushman. Action, outcome, and value: A dual-system framework for morality. *Personality and social psychology review*, 17(3):273–292, 2013.

[29] Pranav Dandekar, Ashish Goel, and David T Lee. Biased assimilation, homophily, and the dynamics of polarization. *Proceedings of the National Academy of Sciences*, 110(15):5791–5796, 2013.

[30] Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*, 2019.

[31] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[32] Stanislas Dehaene, Jean-Pierre Changeux, and Lionel Naccache. The global neuronal workspace model of conscious access: from neuronal architectures to clinical applications. In *Characterizing consciousness: From cognition to the clinic?*, pages 55–84. Springer, 2011.

[33] Anna Dreber, Thomas Pfeiffer, Johan Almenberg, Siri Isaksson, Brad Wilson, Yiling Chen, Brian A Nosek, and Magnus Johannesson. Using prediction markets to estimate the reproducibility of scientific research. *Proceedings of the National Academy of Sciences*, 112(50):15343–15347, 2015.

[34] Sašo Džeroski, Luc De Raedt, and Kurt Driessens. Relational reinforcement learning. *Mach. Learn.*, 43(1-2):7–52, April 2001.

[35] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

[36] P ERDdS and A R&WI. On random graphs i. *Publ. Math. Debrecen*, 6:290–297, 1959.

[37] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.

[38] Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *CoRR*, abs/1711.04574, 2017.

[39] Eugene F Fama. The behavior of stock-market prices. *The journal of Business*, 38(1):34–105, 1965.

[40] Katalin Farkas. Belief may not be a necessary condition for knowledge. *Erkenntnis*, 80(1):185–200, 2015.

[41] Jacques Ferber and Gerhard Weiss. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading, 1999.

[42] Francis Galton. Vox populi (the wisdom of crowds). *Nature*, 75(7):450–451, 1907.

[43] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning, 2016.

[44] John Geweke. Antithetic acceleration of monte carlo integration in bayesian inference. *Journal of Econometrics*, 38(1-2):73–89, 1988.

[45] Antonia Godoy-Lorite, Roger Guimerà, and Marta Sales-Pardo. Long-term evolution of email networks: statistical regularities, predictability and stability of social behaviors. *PloS one*, 11(1):e0146113, 2016.

[46] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[47] Thomas L Griffiths, Charles Kemp, and Joshua B Tenenbaum. Bayesian models of cognition. *The Cambridge Handbook of Computational Psychology, Ron Sun (ed.), Cambridge University Press.*, pages 1–49, 2008.

[48] Thomas L. Griffiths and Joshua B. Tenenbaum. Optimal predictions in everyday cognition. *Psychological Science*, 17(9):767–773, 2006.

[49] Çaglar Gülçehre and Yoshua Bengio. Knowledge matters: Importance of prior information for optimization. *CoRR*, abs/1301.4083, 2013.

[50] John A Hartigan, Pamela M Hartigan, et al. The dip test of unimodality. *The annals of Statistics*, 13(1):70–84, 1985.

[51] Nathan O Hodas and Kristina Lerman. Attention and visibility in an information-rich world. In *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2013.

[52] Aaron B Hoffman and Bob Rehder. The costs of supervised classification: The effect of learning task on conceptual flexibility. *Journal of Experimental Psychology: General*, 139(2):319, 2010.

[53] Achim G Hoffmann et al. General limitations on machine learning. In *ECAI*, pages 345–347, 1990.

[54] Leyla Isik. Perceiving social interactions in the posterior superior temporal sulcus. *Proceedings of the National Academy of Sciences*, 115(1):E113–E114, 2018.

[55] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[56] Daniel Kahneman and Shane Frederick. *Representativeness Revisited: Attribute Substitution in Intuitive Judgment.* Number January 2002. 2014.

[57] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.

[58] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[59] Yea-Seul Kim, Peter Krafft, and Jessica Hullman. Bayesian framework for visualization design and evaluation. Unpublished manusript, 2019.

[60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[61] Max Kleiman-Weiner, Rebecca Saxe, and Joshua B Tenenbaum. Learning a commonsense moral theory. *cognition*, 167:107–123, 2017.

[62] Peter M Krafft, Julia Zheng, Wei Pan, Nicolás Della Penna, Yaniv Altshuler, Erez Shmueli, Joshua B Tenenbaum, and Alex Pentland. Human collective intelligence as distributed bayesian inference. *arXiv preprint arXiv:1608.01987*, 2016.

[63] Peter M Krafft, Julia Zheng, Wei Pan, Nicolás Della Penna, Yaniv Altshuler, Erez Shmueli, Joshua B Tenenbaum, and Alex Pentland. Human collective intelligence as distributed bayesian inference. *arXiv preprint arXiv:1608.01987*, 2016.

[64] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683, 2016.

[65] 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.

[66] David Lazer and Allan Friedman. The network structure of exploration and exploitation. *Administrative Science Quarterly*, 52(4):667–694, 2007.

[67] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Computational social science. *Science*, 323(5915):721–723, 2009.

[68] J. Lorenz, H. Rauhut, F. Schweitzer, and D. Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020–9025, 2011.

[69] Sergio Valcarcel Macua, Aleksi Tukiainen, Daniel García-Ocaña Hernández, David Baldazo, Enrique Munoz de Cote, and Santiago Zazo. Diff-dac: Distributed actor-critic for multitask deep reinforcement learning. *arXiv preprint arXiv:1710.10363*, 2017.

[70] Gabriel Madirolas and Gonzalo G de Polavieja. Improving collective estimations using resistance to social influence. *PLoS computational biology*, 11(11):e1004594, 2015.

[71] Burton G Malkiel and Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

[72] Harry Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.

[73] Giovanna Miritello, Rubén Lara, Manuel Cebrian, and Esteban Moro. Limited communication capacity unveils strategies for human interaction. *Scientific reports*, 3:1950, 2013.

[74] Giovanna Miritello, Esteban Moro, Rubén Lara, Rocío Martínez-López, John Belchamber, Sam GB Roberts, and Robin IM Dunbar. Time as a limited resource: Communication strategy in mobile phone networks. *Social Networks*, 35(1):89–95, 2013.

[75] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. pages 1928–1937, 2016.

[76] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[77] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[78] Mehdi Moussaïd, Juliane E Kämmer, Pantelis P Analytis, and Hansjörg Neth. Social influence and the collective dynamics of opinion formation. *PloS one*, 8(11):e78433, 2013.

[79] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alcssandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015.

[80] Angelia Nedic. Asynchronous broadcast-based convex optimization over a network. *IEEE Transactions on Automatic Control*, 56(6):1337–1351, 2011.

[81] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *arXiv preprint arXiv:1709.08765*, 2017.

[82] Angelia Nedic and Asuman Ozdaglar. 10 cooperative distributed multi-agent. *Convex Optimization in Signal Processing and Communications*, 340, 2010.

[83] Salih N Neftci. Naive trading rules in financial markets and wiener-kolmogorov prediction theory: a study of" technical analysis". *Journal of Business*, pages 549–571, 1991.

[84] Mark Newman. Networks: An introduction, 2010.

[85] Michael Nofer and Oliver Hinz. Are crowds on the internet wiser than experts? the case of a stock prediction community. *Journal of Business Economics*, 84(3):303–338, 2014.

[86] Alejandro Noriega-Campero, Abdullah Almaatouq, Peter Krafft, Abdulrahman Alotaibi, Mehdi Moussaid, and Alex Pentland. The wisdom of the network: How adaptive networks promote collective intelligence. *arXiv preprint arXiv:1805.04766*, 2018.

[87] Andreas Oehler, Matthias Horn, and Stefan Wendt. Brexit: Short-term stock price effects and the impact of firm-level internationalization. *Finance Research Letters*, 22:175–181, 2017.

[88] OpenAI. Roboschool. `https://github.com/openai/roboschool`, 2017. Accessed: 2017-09-30.

[89] Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks, 2017.

[90] Wei Pan et al. *Reality hedging: social system approach for understanding economic and financial dynamics.* PhD thesis, Massachusetts Institute of Technology, 2015.

[91] Cheol-Ho Park and Scott H Irwin. What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, 21(4):786–826, 2007.

[92] Judea Pearl. *Causality.* Cambridge university press, 2009.

[93] Alexander Peysakhovich. Reinforcement learning and inverse reinforcement learning with system 1 and system 2. *arXiv preprint arXiv:1811.08549*, 2018.

[94] Neil C Rabinowitz, Frank Perbet, H Francis Song, Chiyuan Zhang, SM Eslami, and Matthew Botvinick. Machine theory of mind. *arXiv preprint arXiv:1802.07740*, 2018.

[95] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79, 1999.

[96] Ingo Rechenberg. Evolution strategy: Optimization of technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart*, 104:15–16, 1973.

[97] Hilary Richardson, Grace Lisandrelli, Alexa Riobueno-Naylor, and Rebecca Saxe. Development of the social brain from age three to twelve years. *Nature Communications*, 9(1):1–12, 2018.

[98] Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving, 2017.

[99] Daniel M Romero, Brian Uzzi, and Jon Kleinberg. Social networks under stress. In *Proceedings of the 25th International Conference on World Wide Web*, pages 9–20. International World Wide Web Conferences Steering Committee, 2016.

[100] Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[101] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning, 2017.

[102] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: mit einer vergleichenden Einführung in die Hill-Climbing-und Zufallsstrategie.* Birkhäuser, 1977.

[103] John R Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424, 1980.

[104] Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge, 2016.

[105] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[106] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[107] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012.

[108] Giulio Tononi, Melanie Boly, Marcello Massimini, and Christof Koch. Integrated information theory: from consciousness to its physical substrate. *Nature Reviews Neuroscience*, 17(7):450, 2016.

[109] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. In *Social Networks*, pages 179–197. Elsevier, 1977.

[110] Brandon M Turner, Mark Steyvers, Edgar C Merkle, David V Budescu, and Thomas S Wallsten. Forecast aggregation via recalibration. *Machine learning*, 95(3):261–289, 2014.

[111] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131, 1974.

[112] Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.

[113] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[114] Edward Vul and Harold Pashler. Measuring the crowd within probabilistic representations within individuals. *Psychological Science*, 19(7):645–647, 2008.

[115] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[116] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

[117] David H Wolpert and Kagan Tumer. An introduction to collective intelligence. *arXiv preprint cs/9908014*, 1999.

[118] Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W Malone. Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688, 2010.

[119] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, and Peter Battaglia. Relational deep reinforcement learning, 2018.

[120] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757*, 2018.