University of California
Santa Barbara

Modeling and Control of Large Scale Neural Systems

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Brian Mitchell

Committee in charge:

Professor Linda Petzold, Chair
Professor Scott Grafton
Professor Xifeng Yan

March 2020

The Dissertation of Brian Mitchell is approved.

Professor Scott Grafton

Professor Xifeng Yan

Professor Linda Petzold, Committee Chair

December 2019

Modeling and Control of Large Scale Neural Systems

by

Brian Mitchell

Acknowledgements

It should go without saying that this body of work would not exist without the community of people who have been supportive of me while I was building it. But given the opportunity, who could pass up a chance to thank the people who helped bring it into existence?

My committee was most directly involved with and supportive of the work created. The innumerable brainstorming sessions, help with refining my results, and guiding of their presentation helped to take a vague set of ideas and transform them into knowledge. I am extremely grateful and hopefully, other people will benefit as well from their efforts.

My partner and wife Aubrey Jordan supported me every step of the way: from the decision to get my Ph.D. in the first place, all the way through the moment when I wrote these words of thanks. Every late night I spent working was time I could have been spending with her, all the time she spent taking care of me after a long session of work was time she could have been spending on herself. I can only hope that the rewards that come from the path this degree has put me on will help to repay her for all her efforts.

Even though she probably could have been, Aubrey wasn't alone in her support of me. Both of our families contributed their time to keep us afloat and sane during this degree. We would both be far less happy and successful people without their help.

And finally, I'd like to thank my dog. Since he seems to get smarter everyday, I'm holding out hope that he'll be able to read this one day. I highly recommend all prospective graduate students get one during their degree. Just by existing and having needs, he forced me to take regular breaks from work and reminded me to enjoy simple things (e.g. the smell of every single meal I ate).

Curriculum Vitæ
Brian Mitchell

Education

2019	Ph.D. in Computer Science (Expected), University of California, Santa Barbara.
2014	M.S. in Computer Science, University of Chicago.
2009	B.A. in Neuroscience, The Johns Hopkins University

Publications

Mitchell, B.A., Wymbs, N., Grafton, S.T., and Petzold, L.R. A minimum free energy model for network structured systems. Submitted to Proceedings of the National Academy of Sciences.

Mitchell, B.A., Dundon, N., Grafton, S.T., and Petzold, L.R. Inverse Reinforcement Learning in Multi-Objective Environments. Submitted to The Journal of Machine Learning Research.

Mitchell, B.A., Marneweck, M.M., Grafton, S.T., and Petzold, L.R. Motor Adaptation via Distributional Learning. Submitted to The Journal of Neural Engineering.

Mitchell, B.A., Lauharatanahirun, N., Garcia, J.O., Wymbs, N., Grafton, S.T., Vettel, J.M., Petzold, L.R. (2019). A minimum free energy model of motor learning. Neural Computation.

Mitchell, B.A. and Petzold, L.R. (2018). Control of neural systems at multiple scales using model-free, deep reinforcement learning. Scientific Reports.

Abstract

Modeling and Control of Large Scale Neural Systems

by

Brian Mitchell

The control of large scale neural systems promises to usher in a new era of technologies for use in treating disease, answering scientific questions, and improving human performance. Unlike other systems amenable to modern control and reinforcement learning (e.g. robots), we have many examples of high-functioning brains being used by healthy individuals. Combined with the fact that exploratory perturbations of the brain are extremely expensive relative to observation of healthy brains, this suggests that an appropriate workflow for constructing a controller should begin with inverse methods. We present a number of results showing how inverse schemes can be used to both model human behavior and neural dynamics. We show how these methods can be used to facilitate interactions between human beings and artificial agents. We conclude by suggesting that with further refinement, such schemes can be used to directly guide perturbation of the brain.

Contents

Curriculum Vitae **v**

Abstract **vi**

1 Introduction and Background **1**
1.1 Large-Scale Neural Control 2
1.2 Reinforcement Learning 5

2 Deep Reinforcement Learning and Neural Control **11**
2.1 Network of Stochastic, Leaky Integrate and Fire Neurons 15
2.2 Kuramoto Model 21
2.3 Discussion 27
2.4 Experimental Details 29

3 Minimum Free Energy Optimization and Motor Learning **31**
3.1 Data Collection 34
3.2 Behavioral Variability Persists During Motor Learning 41
3.3 Motor Learning Follows Fokker-Planck Dynamics 42
3.4 Fokker-Planck Dynamics are Generated via Free Energy Optimization . . 48
3.5 Each Subject Learns the Same Optimal Behavior 49
3.6 Discussion 52

4 Distributional Temporal Difference Models **56**
4.1 Data Collection 59
4.2 Errors During Motor Learning are Probabilistically Distributed 67
4.3 A Distributional Model for Prediction Errors 73
4.4 Global Neural Activity Optimizes the Distributional Temporal Difference Objective 76
4.5 Robots Can Optimize the Distributional Temporal Difference Objective . 82
4.6 Discussion 86

5 Human Learning and Multi-Objective Reinforcement Learning **89**
5.1 Multi-Objective Inverse Reinforcement Learning 92
5.2 Experimental Validation 96

6 Network Minimum Free Energy Learning **106**
6.1 Data Collection 109
6.2 Network Minimum Free Energy Learning 110
6.3 NMFEL Learning Paths 112
6.4 Local Influence on Global Dynamics 117
6.5 Learning Algorithm Selection 118
6.6 Discussion 122

7 Conclusion and Future Directions **124**

Bibliography **127**

Chapter 1

Introduction and Background

The success of technologies like Trans-Cranial Direct Current Stimulation (TDCS) and Trans-Cranial Magnetic Stimulation (TMS) has marked the beginning of a new era in neuroscience. The ability to safely perturb the brain and induce predictable behavioral change has and will continue to change the way we think about the relationship between the brain, behavior, and technology. The use of such tools is still in its infancy though. In particular, these technologies are known to have global effects on neural activity despite the fact that they are applied to small, focal areas of the brain. The selection of these stimulation sites is largely based on work relating the activity of individual brain regions to behavior. In addition, the longitudinal effects of these technologies is largely unknown. We argue in this thesis that new methods are required to make better use of these technologies: these methods should relate changes in global activity to behavior over time. We are not the first to argue for this need, but we believe that current approaches can be augmented in two primary ways:

1. There is a need for data-driven, dynamical models of neural activity. The literature on large-scale neural systems contains many works with state-of-the-art statistics, but

there has been far less work on modeling systems in a manner amenable to control.

2. Global models of neural activity must be consistent with state-of-the-art approaches to the control of artificial systems. In order to facilitate the interaction of the brain with external devices which also must be controlled (e.g. prosthetics or virtual environments), it is important to think of neural modeling and control within the same frameworks used for these devices.

In this thesis, we propose to approach the problem of large-scale neural control using modern reinforcement learning. This approach has a number of advantages, but presents one major difficulty, namely the design of an appropriate optimization objective. The rest of this thesis presents a number of ways in which this problem can be overcome. We extend methods from inverse reinforcement and imitation learning to show how the objectives optimized by the human brain can be recovered and studied. Consistent with our two objectives above, these approaches yield new, data-driven, global models of neural activity. Further, we show that the optimization objectives recovered can also be used to train artificial agents to reproduce human behavior.

In the rest of this section, we give the background necessary to understand the results presented in subsequent sections. We conclude this thesis by discussing a number of directions in which our methods might be used and extended.

1.1 Large-Scale Neural Control

Past approaches to neural control have focused on dynamical systems-based formulations of the control problem, where methods have been designed to control a single model of

neural dynamics [1, 2, 3, 4, 5, 6, 7, 8]. Specifically, in [7] a method is developed for directly optimizing oscillator coupling strength to impose synchronization on the system. While an interesting approach, this method is unlikely to directly lead to algorithms that can be implemented on real neural systems because the coupling weights between oscillators can be difficult to perturb. The authors of [5, 6] assume that a common forcing input is applied to all oscillators, limiting its applicability to systems where multiple, independent actuators are present. In addition, these control models are open-loop strategies: the authors give a number of motivations for this approach, stating that system dynamics and network connectivity is difficult to estimate and state information may be unavailable. For large scale neural systems, while the dynamics are largely unknown, high spatial resolution state observations can be obtained using functional Magnetic Resonance Imaging (fMRI), and detailed network connectivity can be obtained using Diffusion Tensor MRI (DT-MRI). More importantly though, while their results show that this forcing term can lower the minimum coupling strength at which synchronization occurs, there is still a lower-bound below which synchronization no longer occurs, even in the presence of forcing.

Other dynamical systems approaches include [1, 2, 3, 4, 8], where the authors employ phase reduction methods to models of neural oscillations to better understand how one might desynchronize [1, 2, 3, 4] or synchronize [8] the oscillators (though in [8], the problem of control was mentioned as a potential application, but not studied). These methods involve a search for a reduced set of phases (often a single scalar), whose dynamics well-characterize the oscillatory behavior of each oscillator in the network. The study of the movement of oscillators towards (synchronization) or away from (desynchronization) limit cycles is the focus of these papers. These methods are different from ours, not only in the fact that they are model-based, but also because they work primarily

with low-dimensional dynamics. In particular, it is not clear how these methods would apply to similar control problems where the dynamics are high-dimensional.

Statistical approaches to control have been explored, with both model-based [9, 10] and model-free variants [11, 12] having been attempted. In particular, the authors of [10] show how a controller based on a Generalized Linear Model (GLM) can be used to produce target spike sequences in an underactuated setting. While impressive, it is not clear how these results extend to the ability to generate or manipulate biologically meaningful states. Relying on correlated activity between neurons, in Chapter 2 we show how principal component trajectories can be induced in an underactuated setting. Principal Component Analysis (PCA) has been used in a number of experiments attempting to reduce neural dynamics to a lower dimensional manifold [13, 14, 15, 16]. The dynamics in the phase space defined by a few principal components has been related to simple behaviors. For example, reaching a specific point in space results in a characteristic trajectory of the primary motor cortex in the phase space of its first three principal components. And while the authors of [11, 12] also employ model-free reinforcement learning for neural control, they attempt to solve a different problem from the ones we consider in this thesis: that of the desynchronization of a network of synchronized, coupled oscillators. The advancements of deep reinforcement learning since the publication of these methods suggest that model-free reinforcement learning might be applied to more difficult, poorly understood problems.

1.2 Reinforcement Learning

Reinforcement learning problems are often defined within the framework of a Markov Decision Process (MDP). We similarly adopt this framework in this work. An MDP is defined by the tuple $(\mathbb{S}, \mathbb{A}, p, r, \gamma, \rho_0, H)$. Here $\mathbb{S}$ is the state space, $\mathbb{A}$ is the action space, $p(s^{'}|s, a)$ is the environment dynamics, $r : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}$ is a reward function, ρ_0 is the initial state distribution, $\gamma \in \mathbb{R}$ is a discount factor, and $H \in \mathbb{R}$ is the horizon.

1.2.1 Policy Optimization in Reinforcement Learning

One of the problems that can be solved using RL in a MDP is policy optimization, where a policy, $\pi_\theta(a|s)$, is a function parameterized by θ which accepts a state and outputs an action. The parameters of π can be fit by solving the optimization problem

$$\max_{\theta} \mathrm{E}_\tau[\mathcal{L}(\tau|\theta)], \tag{1.1}$$

where $\tau = \{(s_{t'}, a_{t'}), \ldots, (s_{t'+T}, a_{t'+T})\}$ is a trajectory of state-action pairs, T is a time horizon, and E_τ is the expectation operator over trajectories generated from $\pi_\theta(a|s)$ and $p(s'|s, a)$. $\mathcal{L}$ is an objective function characterizing the performance of the policy over a trajectory. Here, we use the sum of future rewards to define $\mathcal{L}(\tau|\theta) = \sum_{t=t'}^{t'+T} r(s_t, a_t)$. A common strategy to solving the optimization problem in Equation 1.1 is using gradient descent:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta \mathrm{E}_\tau[\mathcal{L}(\tau|\theta_k)]$$

,

where $\alpha \in \mathbb{R}$ is a step size parameter. The update can be rewritten more generally as

$$\theta_{k+1} = \theta_k + w_\psi(\nabla_\theta \mathrm{E}[\mathcal{L}(\tau|\theta)]), \tag{1.2}$$

where w_ψ is the update function parameterized by ψ. Various ways of defining w as well as fitting ψ have been used in the meta-learning literature [17, 18, 19]. Approaches have been developed to deal with non-stationary environments with single reward functions or multi-task environments [18, 20]. In these cases, models are sought that are able to quickly update an existing policy to new tasks or a changing environment, or to initialize a naive policy such that it learns a new task quickly. The success of a meta-learner is based on its ability to produce good performance on an arbitrary new task characterized by a scalar reward function. In this thesis, we use expert demonstrations to ensure good performance on a single multi-objective problem, rather than fast learning of new single-objective problems. Our work is similar to these approaches though, in the sense that we make use of the same kind of general gradient-based update used in meta-learning methods.

1.2.2 Deep Q-Learning

The goal of Q-learning is to estimate an action-value function (i.e. a Q-function) which is a measure of the expected future rewards, given a fixed policy, $\mu(a_t|s_t)$. This Q-function may then be used for policy optimization. One of the particularly attractive features of Q-learning is that it is a model-free control strategy. This means that no explicit model of the state-transition dynamics is estimated during computation of the policy. Thus for Q-learning, particular importance is placed on finding a good estimator of the Q-function. In Deep Q-Learning, a deep neural network is used to approximate Q.

The MDP framework allows for a convenient representation of the Q-function. The Q-function can be written as

$$Q^{\mu}(a_t, s_t) = \mathrm{E}_{r_{i\geq t}, s_{t>t}, a_{i>t}\sim\mu}[r_t|a_t, s_t], \tag{1.3}$$

where $r_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$, $\gamma \in \mathbb{R}$ is the discounting factor, and $a \sim \mu$ indicates the sampling of an action from the policy. Under an MDP, it may be written as

$$Q^{\mu}(a_t, s_t) = \mathrm{E}_{r_t, s_{t+1}}[r(s_t, a_t) + \gamma \mathrm{E}_{a_{t+1}\sim\mu}[Q^{\mu}(a_{t+1}, s_{t+1})]], \tag{1.4}$$

where this equation is known as the Bellman equation. A common approach to estimating the optimal policy from the Bellman equation is to estimate μ in a greedy fashion by computing $\mu(a_t|s_t) = \arg\max_{a_t} Q^{\mu}(a_t, s_t)$. This is the approach taken in [21], where a deep convolutional network is used to approximate the Q-function.

While this approach has been extremely successful on a number of different problems, it is impractical for large, continuous action spaces. Lillicrap et al. [22] proposed to modify this approach by incorporating a Deterministic Policy Gradient (DPG) (graphical description in Figure 2.1) [23]. In this case, μ is assumed to be a deterministic function and the parameters of the policy are updated along the following gradient

$$\nabla_{\theta^{\mu}} J \approx \mathrm{E}_{s_t\sim\rho^{\beta}}[\nabla_{\theta^{\mu}} Q^{\mu}(a, s|\theta^{Q})|_{s=s_t, a=\mu(s_t|\theta^{\mu})}], \tag{1.5}$$

$$\nabla_{\theta^{\mu}} J \approx \mathrm{E}_{s_t\sim\rho^{\beta}}[\nabla_a Q^{\mu}(a, s|\theta^{Q})|_{s=s_t, a=\mu(s_t|\theta^{\mu})} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})_{s=s_t}], \tag{1.6}$$

where ρ^{β} is a behavior policy potentially distinct from μ, and θ^{μ} and θ^{Q} are the parameters of the policy and the value function respectively. The utility of an off-policy algorithm in

the context of neural control (and biological control in general) is significant. For the systems considered in this thesis, being able to estimate the policy gradient in an off-policy fashion allows us to avoid resampling states with every step of the gradient optimization and to reuse past samples (this is known as experience replay). In the setting of general biological control, sampling from a policy could involve a procedure with a deleterious cumulative effect (e.g. stimulating a collection of neurons as in deep brain stimulation) and the ability to minimize the number of times this is performed could be quite valuable.

The parameters θ^Q are updated along the gradient of a separate loss function

$$L(\theta^Q) = \mathrm{E}_{s_t \sim \rho^\beta, a_t \sim \beta}[(Q^\mu(a_t, s_t|\theta^Q) - y_t)^2], \tag{1.7}$$

where y_t is given by

$$y_t = r(s_t, a_t) + \gamma Q^\mu(a_{t+1}, s_{t+1}|\theta^Q). \tag{1.8}$$

The loss function $L(\theta^Q)$ is known as the temporal difference error (TD-error) and there is a long literature about its properties and potential applications [24]. The gradients, $\nabla_{\theta^\mu} J$ and $\nabla_{\theta^Q} L$, are used to alternately perform updates of θ^μ and θ^Q respectively. This approach is termed an actor-critic method, where the actor, $\mu(a_t|s_t, \theta^\mu)$, is used to propose actions and the critic, $Q^\mu(a_t, s_t|\theta^Q)$, declares the value of those actions. In this thesis, we directly use DDPG to control simulated large-scale neural systems. We also use a distributional form of Q-Learning to model the optimization objective of large-scale neural systemss.

1.2.3 Multi-Objective Optimization

Multi-objective optimization addresses problems where multiple objectives must be simultaneously optimized. If these objectives are independent, this problem can be addressed straightforwardly by optimizing the objectives in sequence. Unfortunately, this is rarely the case in practice. The existing literature focuses on exploiting various kinds of correlation structure across the objectives to achieve an acceptable compromise across them. For example, one popular approach is known as reward sharing, where a collection of reward functions are all highly correlated with a single reward function [25, 19, 20], and this single reward is fit using a latent variable model. If this reward function can be found and a map can be found from the multi-objective space to the single reward space, then the multi-objective problem can be effectively reduced to a conventional RL problem. A related approach is reward scalarization, which involves mapping the multiple rewards to a single reward [26, 27, 28]. In this work, we don't assume the ability to perform dimensionality reduction or scalarization, but rather attempt to learn to move optimally through the multivariate objective space.

More formally, we assume that we have n performance objectives to use in updating our policy, $\mathrm{E}[\mathcal{L}_1], \ldots, \mathrm{E}[\mathcal{L}_n]$. The notion of optimality of a policy in the presence of n objectives can be defined in terms of a Pareto set of policy parameters. Specifically, the Pareto set $\mathcal{P}$ can be defined as the set of non-dominated policy parameters, where θ^* dominates θ if $\mathrm{E}[\mathcal{L}_i(\tau|\theta^*)] \geq \mathrm{E}[\mathcal{L}_i(\tau|\theta)]$, for all $1 \leq i \leq n$ and there exists at least one i for which $\mathrm{E}[\mathcal{L}_i(\tau|\theta^*)] > \mathrm{E}[\mathcal{L}_i(\tau|\theta)]$. Computation of $\mathcal{P}$ is slightly different than optimization in the single objective setting. Specifically, a primary goal of Equation 1.1 is to find θ such that the necessary optimality condition, $\nabla_\theta \mathrm{E}[\mathcal{L}(\tau|\theta)] = \mathbf{0}$, where $\mathbf{0}$ is a vector of zeros, is approximately satisfied. This is not necessarily possible in the case

where negative correlations exist across multiple objectives. Instead, a different necessary condition for optimality is used. In particular, we note that if θ^* is an element of $\mathcal{P}$, then

$$\sum_i \alpha_i \nabla_\theta \mathrm{E}[\mathcal{L}_i(\tau|\theta^*)] = \mathbf{0}, \tag{1.9}$$

where $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ are all non-negative and $\sum_i \alpha_i = 1$. To find a vector $\alpha = (\alpha_1, \ldots, \alpha_n)$ that satisfies Equation 1.9, it is sufficient to explicitly minimize $||\sum_i \alpha_i \nabla_\theta \mathrm{E}[\mathcal{L}_i(\tau|\theta^*)]||_2^2$ with respect to α. In fact, if α^* is a solution to this optimization problem, then $-\sum_i \alpha_i^* \nabla_\theta \mathrm{E}[\mathcal{L}_i(\tau|\theta^*)]$ is a valid descent direction on the objectives $\mathrm{E}[\mathcal{L}_1], \ldots, \mathrm{E}[\mathcal{L}_n]$. Interestingly, this convex combination of the gradients of our objectives is simply a particular choice of w_ψ in the multiobjective setting. This connection allows us to rewrite Equation 1.2 as

$$\theta_{k+1} = \theta_k + w_\psi(\nabla_\theta \mathrm{E}[\mathcal{L}_1(\tau|\theta_k)], \ldots, \nabla_\theta \mathrm{E}[\mathcal{L}_n(\tau|\theta_k)]). \tag{1.10}$$

Equation 1.10 is a generic gradient-based update in the multiobjective setting. The optimal θ, given by θ^*, is determined by both ψ and the different objectives, $\mathrm{E}[\mathcal{L}_1], \ldots, \mathrm{E}[\mathcal{L}_n]$. We use this information in this thesis to formulate the problem of Multi-Objective Inverse Reinforcement Learning (MOIRL) and then to devise an algorithm to solve it.

Chapter 2

Deep Reinforcement Learning and Neural Control

With the advent of deep learning and deep neural networks, reinforcement learning has rapidly advanced beyond methods in classical neuro-dynamic programming by the publication of the Deep Q-Network (DQN) [21]. This approach to Q-learning relies on convolutional neural network approximations of both the policy as well as the value (Q) function and was used to allow a computer to better learn to play Atari games. Since its publication, deep Q-learning has been applied to a number of different physical problems. Particularly relevant to neural control is the extension of DQN to problems with continuous state and action spaces by the Deep Deterministic Policy Gradient (DDPG) method [22].

In this chapter, we describe the first application of deep reinforcement learning for control of biological systems [29]. There are many reasons why this has not been attempted before the work in [29], but perhaps the most important and serious hurdle involves the difficulty in constructing accurate models of complex biological systems amenable to

control. This is not to say that good models don't exist: in fact, a number of approaches have been tried including dynamical systems [1, 2, 3, 4, 5, 6, 7, 8] and statistical approaches using a point-process model of past inputs and spikes [9, 10, 11, 12, 16]. It is certainly possible to incorporate a deep neural network into these existing approaches (e.g. using a deep neural network to fit the rate of the point process, as in [30]). From either a dynamical systems or a statistical perspective though, networks of neurons are difficult to model in that the modeling process requires significant interdisciplinary insight. In addition to the non-linearity and stochasticity of these systems, there is the added difficulty of selecting the scale at which to model these systems. Behavior can be strongly influenced by single protein dynamics as well as small and large neural population dynamics and the selection of the scale at which to model these systems can be non-trivial. Moreover, the construction of a model for a single scale is typically useless when trying to model events at other scales.

Due to the current complexities and limitations of models of actual neural systems, we have chosen to pursue a control strategy that is model-free (i.e. DDPG). This framework frees us from the need to model the state transition dynamics at all. Such an approach may be useful for fields including translational research whose goal is to generate medical therapies: in these areas, understanding the dynamics of the pathological system is often thought of as a prerequisite for developing a therapy. We argue that our results indicate that this may not always be necessary. These results would also allow researchers to easily adapt our control system to changes in system dynamics. While these advantages are significant, there are additional challenges posed by a model-free strategy that are not present in a model-based control strategy. For example, many have criticized model-free methods for their poor sample complexity [31], claiming that this limits their applicability to real systems [32]. We observe that for the synthetic neural

systems used in this chapter, a Kuramoto Model (KM) of synchronized oscillators and a network of Stochastic, Leaky Integrate and Fire (SLIF) neurons, the sample complexity was reasonable given proper definition of the reward function.

This point raises another of the difficulties involved in model-free control, namely the proper design of a reward function. The reward function must be precisely designed so as to inform the policy as to the goals of the problem, but the state-action space with non-zero reward must be reachable in a reasonable amount of time from the initial conditions. This relationship requires proper coordination between the design of the reward function, the definition of the state space, and the use of exploration during fitting of the policy and value functions. For SLIF control (both fully- and under-actuated problems), we find that simple rewards are sufficient to achieve good performance. Though for control of the Kuramoto Model, we make use of shaped rewards. These reward functions allow for the decomposition of complex tasks into simpler ones with smaller rewards being given for partial solutions of the full objective. In this case, the full objective we consider is the synchronization of a network of weakly coupled oscillators with the intermediate objective of synchronization of each oscillator in the network with a reference oscillator (i.e. entrainment of the network). In addition, we make use of an Ornstein-Uhlenbeck process for the exploration of the state-action space. This choice was motivated both by the previous use of this process with DDPG as well as the neurological significance of this process. For example, under certain conditions it is thought that the evolution of synaptic spine strength follows an Ornstein-Uhlenbeck process [33, 34]. Since the connection strength between two cells is roughly proportional to the sizes of the spines on the synapes between them, exploration of the space of spine sizes can be interpreted as exporing the space of possible Extra-cellular Post-Synaptic Potentials that may be generated by a given pre-synaptic cell. Further, Ornstein-Uhlenbeck noise has also recently

been used as a model of large-scale neural dynamics [35] during a decision making task.

We claim that DDPG, and deep reinforcement learning in general, has considerable promise for general purpose neural control. We describe the models that are the object of the control and the manner in which the problem is formulated for each model system. For each system, the objective is slightly different, the analysis must be slightly different. In general, we show empirically that we are able to achieve a wide array of different objectives on each system in a sample efficient and scalable manner.

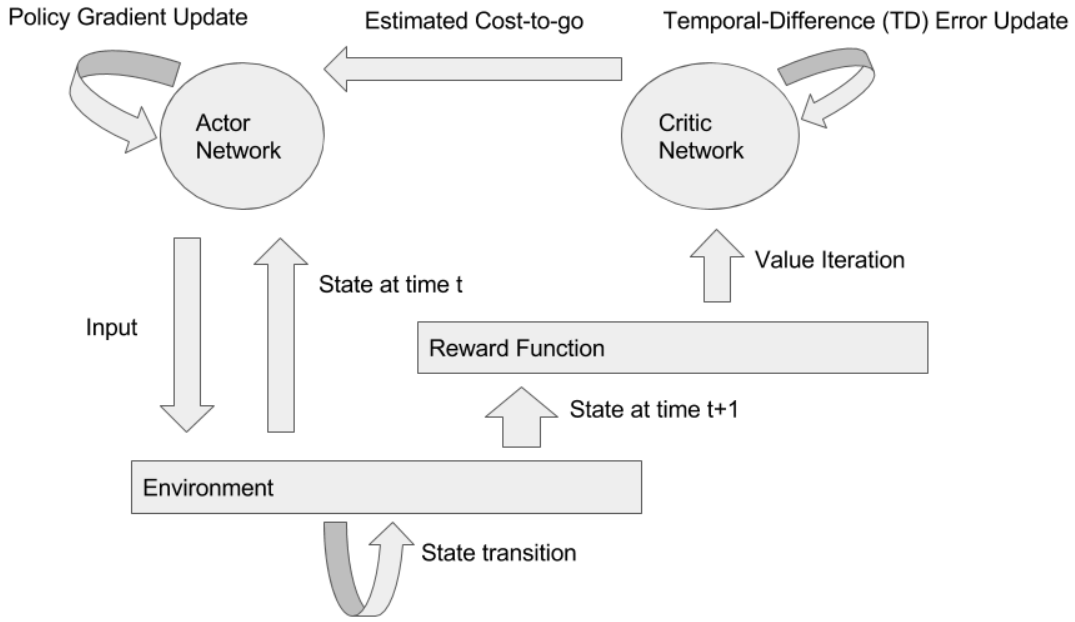


Figure 2.1: Diagram of the DDPG algorithm. The core components of this algorithm are the actor network (i.e. the policy), the critic network (i.e. the value function), and the environment (in this work, either the SLIF or Kuramoto model). The environment accepts inputs from the actor network and produces a state transition. The new state and action used to generate it are passed to the reward function. The reward function passes an update to the critic network, which estimates the value of the state-action pair at the current timestep. The parameters of the actor network are updated using the new policy gradient and the parameters of the critic network are updated using the new TD-error.

2.1 Network of Stochastic, Leaky Integrate and Fire Neurons

A great deal has been published about simulated networks of single neurons, making this class of models an important benchmark for our system. We chose the Stochastic, Leaky Integrate and Fire (SLIF) model for the validation of our system, and connected these neurons in a random, directed network. The SLIF model is given by

$$\frac{dV_i(t)}{dt} = \frac{-1}{\tau_v} V_i(t) + \frac{1}{C}\left(bu_j(t) + \sum_j A_{i,j} I_{syn,j}(t)\right) + \eta e_i(t), \tag{2.1}$$

$$I_{syn,j}(t) = -g_{syn,j}(t)\left(V_i(t) - E_{syn}\right), \tag{2.2}$$

$$g_{syn,j}(t) = \bar{g}\frac{t - t_s}{\tau_s}\exp(\frac{-(t - t_s)}{\tau_s}), \tag{2.3}$$

where τ_v is the membrane time constant, C is the membrane capacitance, $e_i(t)$ is the standard Gaussian white noise of the i'th cell, η denotes the standard deviation of this noise, $u_i(t) \in \mathbb{R}^S$ is the extrinsic control input to the i'th cell, $b \in \mathbb{R}^{1\times S}$ denotes the influence of the input on the neuron, $I_{syn,j}(t)$ is the synaptic current coming from the j'th neuron firing an action potential at time t_s, $A_{i,j}$ is the weight of the connection between the i'th and j'th cells, E_{syn} is the reversal potential of the synapse, $\bar{g}$ models the constant synaptic conductance, and τ_s determines the decay of the synaptic current as time elapses from the incoming spike at t_s. For these experiments, we assume the existence of only a single type of cell, that is, a generic excitatory cell. All values of model parameters used to generate the results shown in this chapter are given in the Experimental Details section.

2.1.1 Fully-Actuated Network Control

For the first application of DDPG, we solve the toy problem of inducing an arbitrary spike train in each neuron of the network, where each target spike train is drawn at random from a Poisson distribution. A model-free controller of this system must learn a number of key tasks: to depolarize each neuron at the appropriate times, to hyperpolarize the neuron at other times, and to generate inputs to each cell sufficient to wash out the effects of neighboring neurons. The simple reward function

$$r(s_{i,t}, a_{i,t}, s_{i,t+1}) = \begin{cases} 1, & \text{if } s_{i,t+1} = h_{i,t+1}, s_{i,t+1} \neq 0 \\ -1, & \text{if } s_{i,t+1} \neq h_{i,t+1} \\ 0, & \text{otherwise} \end{cases}, \tag{2.4}$$

was sufficient to achieve our objective of controlling cells to generate random, independent spike trains. Here, $s_{i,t}$ is the state of the i'th cell at time t and $h_{i,t+1}$ is the target activity of the i'th cell at time $t+1$. In order to achieve control of the i'th cell using DDPG, we generate a separate policy and value function for each cell, resulting in N policies and value functions being estimated. This approach of controlling each cell individually with a separate policy and value function is reasonable for some modern optogenetic and implanted electrode systems where stimulation is capable of overriding the influence of neighboring inputs and the targeting of single cells is possible.

We ran an experiment on a fully-actuated network of 20 SLIF cells where after a 500 episodes, the controller obtains perfect accuracy in inducing target spike trains in each cell in the network. Details regarding the initialization and parameters of this network are presented in the Experimental Details section. One would expect the accuracy of this approach to hold for larger networks if the assumption that stimulation can always

override neighboring inputs holds. Thus in this simulated setting, if not for the larger computational cost incurred by fitting and storing additional policies, controlling a small network is not more difficult than controlling a large network. Unfortunately, there are many other practical issues that make controlling a larger network much more difficult than controlling a smaller network. For example, the field of view through which neurons can be accessed in a live organism typically restricts the amount of hardware that can be used, and thus, the number of neurons that can be simultaneously simulated. One way to incorporate this limitation into our simulation involves the relaxation of the objective of inducing an independent spike train for each cell. We show in the next section how this relaxation can still lead to the induction of biologically significant states in an SLIF network.

2.1.2 Under-Actuated Network Control

In general, it is impossible to induce distinct spike trains in n independent neurons with arbitrarily high accuracy using fewer than n policies. Fortunately, *in vivo* neurons in a network often display highly correlated activity patterns, suggesting that full actuation is not necessary to control many biologically meaningful states. A common model for the correlated behaviors of neurons in a network (and one compatible with the SLIF model) is based on a linear mixture model where the membrane potential of neuron k at time t is given by

$$V_k(t) = \sum_i y_{\text{pre},i}(t) w_i(t) + b_k(t), \tag{2.5}$$

where $y_{\text{pre},i}(t)$ is the post-synaptic potential delivered from neuron i, w_i is the synaptic weight between neurons i and k, and b_k is the change in V_k induced by neuron k itself (accounts for the term $\frac{-1}{\tau_v} V_k(t) + \eta e_k(t)$ in the SLIF equation) [36]. For example, corre-

lated behaviors arise in this model system in the case of fully-connected networks of cells. In this case, if one cell in the cluster generates an action potential, then the change in the membrane potential at a subsequent timestep of all the other members of the cluster will be correlated. The situation where a network is composed of a collection of highly connected communities with sparse inter-community connections results in a covariance matrix that is approximately block-diagonal.

The study of such covariance matrices has a strong literature in the computational neuroscience community, if only indirectly, because of the popularity of methods such as Principal Component Analysis (PCA). For example, the dynamics of many large populations of neurons are known to oscillate on a low-dimensional (e.g. two or three dimensional) manifold (e.g. as shown in [30]). Results demonstrating the low-dimensional structure of the activity of large collections of neurons have been produced for a number of different model systems and conditions [13, 14, 15]. It is known that PCA achieves perfect accuracy in the recovery of communities of neurons interacting via a linear mixture in the case of a covariance matrix with perfect block-diagonal structure [37]. Though, to the best of our knowledge, no previous work exists on controlling principal component trajectories in neural systems. Some recent work exists on spectral control in large scale neural systems [16], but it isn't clear how this work might be applied to neural systems at different scales. We show how an underactuated system can be controlled to induce an arbitrarily structured oscillation in the phase space defined by a small number of principal components.

To do this, we construct an adjacency matrix of network connectivity with an approximate block-diagonal structure and assume recovery of the correct low-dimensional manifold (in general, this is not possible with PCA, but methods like the Treelet Transform

[37] allow the recovery of the correct manifold without requiring perfect block-diagonal structure). Examples of an adjacency matrix with two dimensional dynamics as well as the associated principal components are shown in Figure 2.2. Within each community, we pick a neuron at random to receive input from a distinct policy. We then construct one and two dimensional target oscillations and attempt to reconstruct these oscillations in the phase space defined by the first one or two principal components of the neural activity.

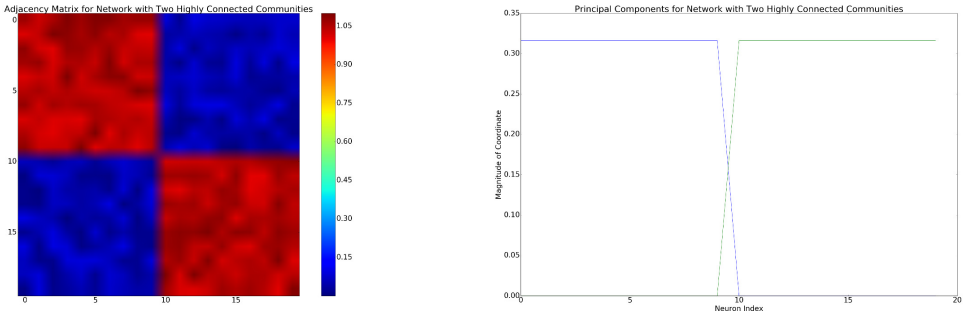


Figure 2.2: **Left**: example of an adjacency matrix with approximate block-diagonal structure. Assuming a linear mixture model of neuronal interactions, this network structure will induce an approximately block diagonal covariance of similar structure. **Right**: the principal components associated with the adjacency matrix on the left.

The reward function we use for accomplishing this is

$$r(\phi_{\text{targ}}(t), \phi_{\text{cntrl}}(t)) = \begin{cases} 1, & \text{if } d(\phi_{\text{targ}}(t), \phi_{\text{cntrl}}(t)) < \epsilon \\ -1, & \text{otherwise} \end{cases}, \tag{2.6}$$

where $\phi_{\text{cntrl}}(t)$ is the phase of the controlled oscillation, $\phi_{\text{targ}}(t)$ is the phase of the target oscillation, $d(\cdot, \cdot)$ is a distance function, and ϵ is a scalar. We define $d(\cdot, \cdot)$ to be 0 when $\phi_{\text{cntrl}}(t)$ is in the correct half of the unit circle at time t and infinity otherwise. To accomplish this, we discretize the unit circle into two halves, $[0, \pi)$ and $[\pi, 2\pi)$; the reward function attempts to force ϕ_{cntrl} to be in the same half as ϕ_{targ} at time t. We use the binary vector of action potentials at time t over all cells in the network to estimate

ϕ_{cntrl} by first projecting it onto the first one or two principal components, then estimating the phase angle. Results from these experiments are shown in Figures 2.3 and 2.4.

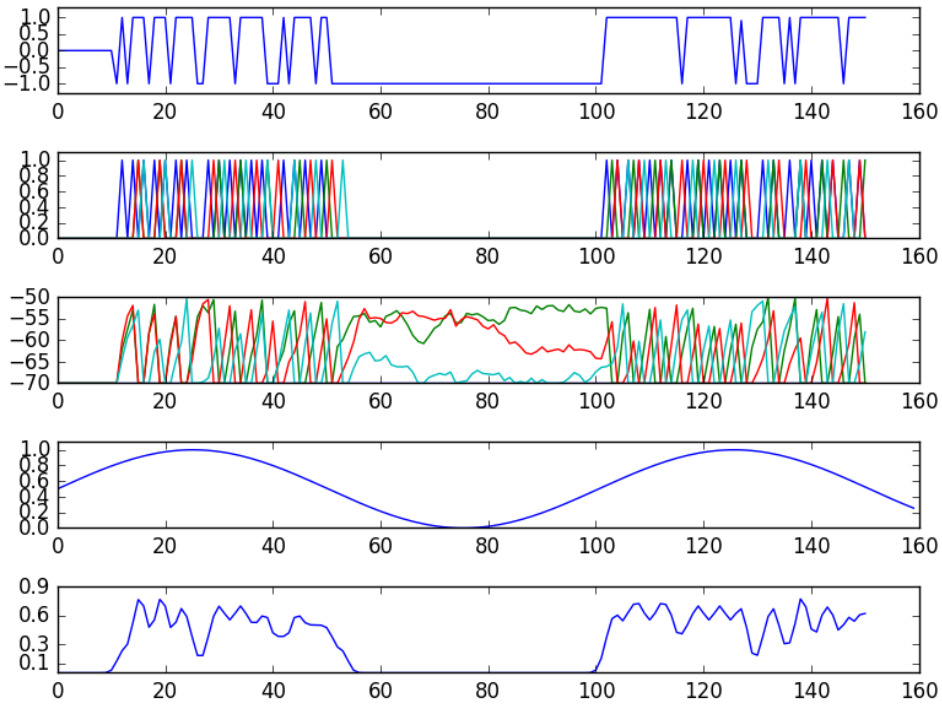


Figure 2.3: Results of the experiment controlling oscillation in the phase space defined by a single principal component. The **first** plot from the top is a plot of the input into the actuated cell over time; the **second** plot from the top is a plot of the spikes of the entire network, where different colors correspond to different cells; the **third** plot from the top corresponds to the membrane potential of each cell over time; the **fourth** from the top plot shows the target oscillation; the **bottom** plot shows the observed oscillation. The policy, despite delivering input to only a single cell, is able to approximately induce the target oscillation in the observed phase space.

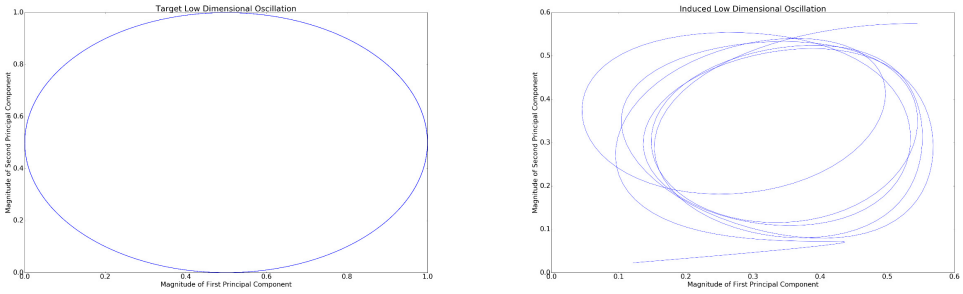


Figure 2.4: In this plot, a two-dimensional oscillation is induced in the phase-space defined by the first two principal components of the network. **Left**: the target oscillation. **Right**: the observed oscillation.

We can see from these results that in the underactuated setting, while the objective is not quite as ambitious as in the fully-actuated setting (i.e. we are not able to induce an arbitrary spike-train in each cell), we can still induce physiologically meaningful states in a reasonably accurate way.

2.2 Kuramoto Model

The Kuramoto Model (KM) characterizes the dynamics of a network of oscillators. These oscillators interact with each other over the network, and the phase of a single oscillator changes in relation to the phases of its neighboring oscillators. The KM is given by

$$\frac{d\phi_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} A_{i,j}\rho(\phi_j - \phi_i), \tag{2.7}$$

where N is the number of oscillators, K is the coupling strength, $A_{i,j}$ is the edge weight between the i'th and j'th oscillators, ϕ_i is the phase of the i'th oscillator, ω_i is the natural frequency of the i'th oscillator, and ρ is a non-linear function characterizing the influence of neighboring oscillators on the i'th oscillator. In the original KM model, $\sin(\cdot)$ was used as the non-linearity. All values of model parameters used to generate the results shown in this chapter are given in the section on Experimental Details.

We adopt a recent control strategy for this model which was originally introduced for disrupting the synchronizing effects of the rightmost term in the KM (e.g. [38, 39]). It is known that for large K, the network of oscillators synchronize over time. An analytical solution for desynchronization of the oscillators is developed in [38]. That framework

takes the form

$$\frac{d\phi_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} A_{i,j}\rho(\phi_j - \phi_i) + \frac{d\phi_i^{\text{ctrl}}}{dt}, \tag{2.8}$$

where ϕ_i^{ctrl} is an additive control term. With DDPG, rather than focusing on the popular problem of desynchronizing oscillators bound to synchronize, we address the problem of inducing synchrony in networks that will not synchronize on their own (i.e. with small K). To relate the controlled KM presented above with the DRL framework, consider a forward Euler discretization of the above ODE:

$$\frac{\phi_{i,t} - \phi_{i,t-1}}{\Delta t} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} A_{i,j}\rho(\phi_{j,t-1} - \phi_{i,t-1}) + \frac{\phi_{i,t}^{\text{ctrl}} - \phi_{i,t-1}^{\text{ctrl}}}{\Delta t}. \tag{2.9}$$

where $\phi_{i,t}^{\text{ctrl}}$ is the input helping induce the transition to $\phi_{i,t+1}$. Solving for $\phi_{i,t}^{\text{ctrl}}$, we obtain

$$\phi_{i,t}^{\text{ctrl}} = -\Delta t\left(\omega_i + \frac{K}{N}\sum_{j=1}^{N} A_{i,j}\rho(\phi_{j,t-1} - \phi_{i,t-1})\right) + \phi_{i,t-1}^{\text{ctrl}} + \phi_{i,t} - \phi_{i,t-1}. \tag{2.10}$$

Here $\phi_{i,t}^{\text{ctrl}}$ can be considered a policy, $\mu_i(a_t^i|s_t)$, where the state, s_t, is $(\phi_{i,t}, \phi_{i,t-1}^{\text{ctrl}}, \phi_{1,t-1}, \ldots, \phi_{N,t-1})$. Using DDPG, we can extend this idea so that $\mu_i(a_t^i|s_t, \theta^{\mu_i})$ can be defined using a deep neural network with parameters θ^{μ_i}.

We apply this approach to the problem of synchronizing a network of weakly coupled oscillators by entrainment to a reference oscillator. This problem has recently been considered in [5, 6]. An important result from this work is that the bound on the coupling strength below which synchronization no longer occurs, K_c^{unf}, is lowered by forcing input to K_c. If $p(\omega)$ is the distribution from which the natural frequencies are drawn and it is unimodal and symmetric about 0, then $K_c^{\text{unf}} = \frac{2}{\pi p(0)}$ as $N \to \infty$. In contrast,

in the presence of forcing, the minimum coupling strength required for synchronization is reduced to $K_c = \frac{2}{\pi}$. Below this coupling strength, while entrainment imposes the reference frequency on each oscillator in a network, the phases achieve nominal synchronization (between 0.0 and 0.2) [5]. To demonstrate the utility of DDPG for model-free control of the Kuramoto model, we pick a coupling strength below K_c ($K = 0.1$) and show that the synchronization achieved is significantly higher than this (about what one would expect for a coupling strength of slightly greater than $\frac{2}{\pi}$ using the method in [5, 6]).

We define the state space to be the set of phases of all oscillators over a 40 timestep history, along with the adjacency matrix of the network. The reward function was defined to be

$$r_i(s_t, a_t^i) = \frac{q + \epsilon q_i' + \eta ||a_t^i||_1}{2 + \epsilon}, \tag{2.11}$$

where $\epsilon, \eta \in [0, 1]$ and q and q_i' are defined to be the order of synchronization. This quantity is given by

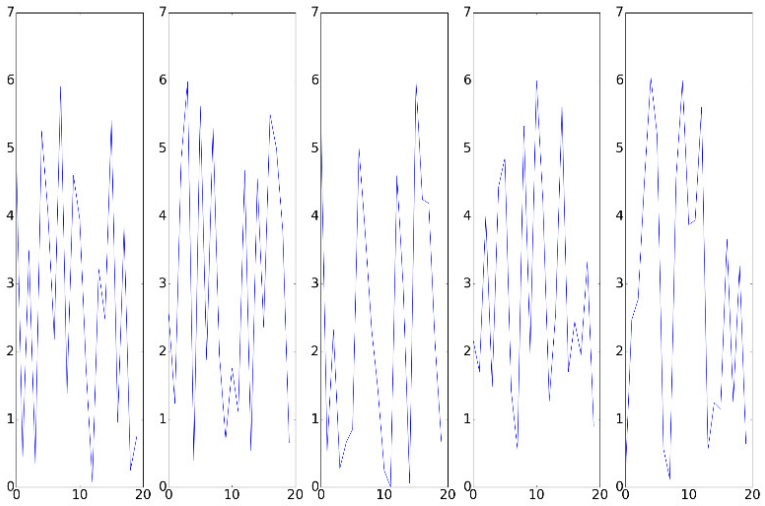
$$q e^{i\psi} = \frac{1}{N} \sum_{j=1}^{N} e^{i\phi_j}, \tag{2.12}$$

where ψ corresponds to the average phase of the oscillators. The difference between q and q' is that q corresponds to the synchronization of all N oscillators, while q_i' is defined to be the synchronization of the i'th oscillator with respect to a reference oscillator. It was observed that without this term in the reward function, the policy regularly failed to induce global synchronization. To stabilize the controller, one oscillator in the network was chosen at random to be the reference, though an external oscillator might also be used. Regularization by the norm of the action was also included, to encourage more reliable exploration of the state-action space. Specifically, without this regularization, the controller would favor large actions, even though these actions were rarely optimal.

This regularization was thus included to encourage more thorough exploration of the state-action space.

It can be seen from Figures 2.5 and 2.6 that the network eventually synchronizes. To understand this result, it may help to view the reward function as a shaped reward. This interpretation follows from the fact that synchronization of each oscillator in the network with a single reference oscillator is far easier than inducing global synchronization of the entire network. This point is emphasized by Figure 2.6 which shows that almost all oscillators in the network achieve very high synchronization with the reference oscillator, and in general, the average synchronization with the reference is much higher than the global synchronization of the entire network. That first synchronizing with the reference makes global synchronization easier to obtain follows from the fact that the size of the state space to be explored is reduced. So, rather than having to tune frequencies and phases of the oscillators in the network, synchronization with the reference frequency-locks all oscillators and thus, global synchrony can be induced by adjusting their respective phases.

a



b

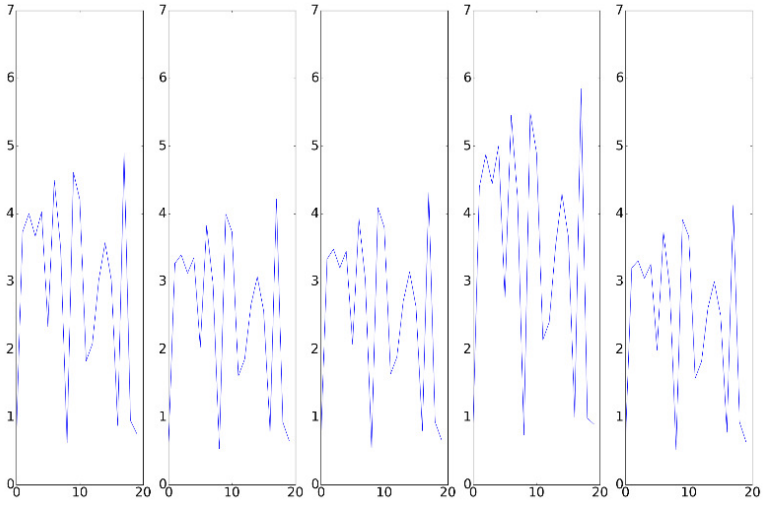
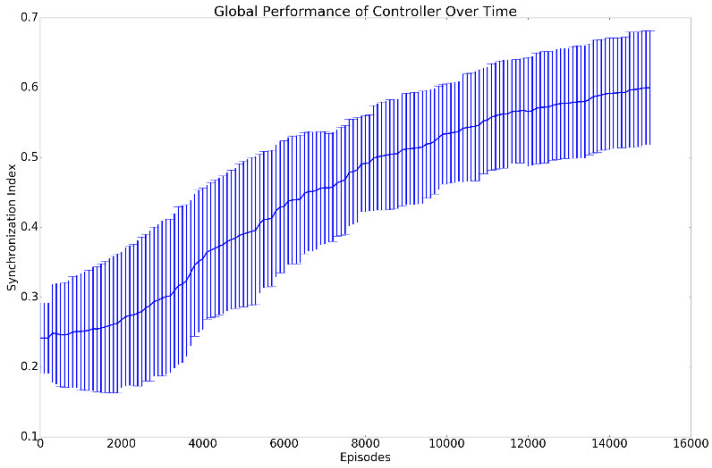


Figure 2.5: Examples demonstrating induction of synchronization by controller for weakly coupled oscillators. For all plots, the vertical axis is the phase and the horizontal axis is timestep. For both sections **a** and **b**, five plots are included generated from five randomly selected oscillators from the network of 20 oscillators. **a**: phases for oscillators before global synchronization has been induced by the controller. **b**: phases for oscillators after global synchronization has been induced by the controller. This level of synchronization can be observed after a few hundred training episodes have been observed.

a



b

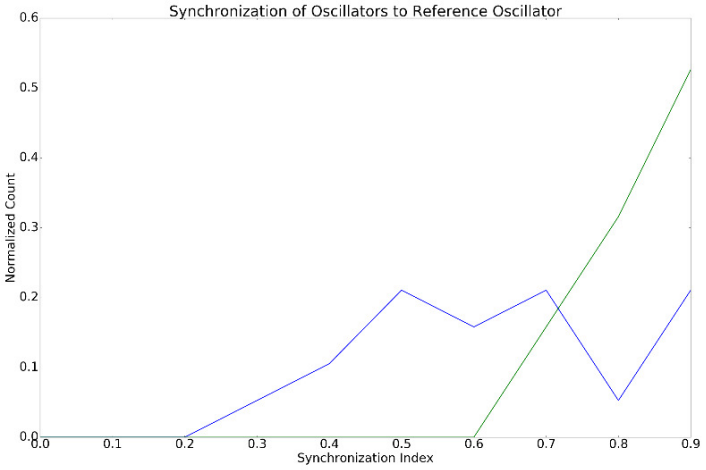


Figure 2.6: Summary results of 10 synchronization experiments. **a** depicts the mean and standard deviation of the global synchronization, (i.e. q from equation 16), against the number of training periods of the controller. **b** shows histograms demonstrating the synchronization level of all network oscillators with the reference oscillator (i.e. q_i from equation 16). That is, a point on either the blue or green curves demonstrates the probability of having a given value for q_i. The **blue** histogram shows counts before training while the **green** histogram shows counts after training. The average synchronization with the reference, q_i, is much higher than global synchronization, q, which is explained by the fact that synchronization with the reference is easier to induce than global synchronization.

2.3 Discussion

The model-free approaches to control of neural systems presented here suggest that deep reinforcement learning has potential for application to this area. We show how the engineering problem is transformed from one that focuses on the design of appropriate system dynamics and the control of these models, to the design of good reward functions that allow for accurate and tractable optimization of the respective objectives. In the case of inducing target spike trains or latent trajectories in the SLIF model, the rewards were able to directly represent these objectives. The problem of inducing synchronization in a weakly coupled network of oscillators required the introduction of a shaped reward to allow for robust synchronization of the network. The problem of designing appropriately shaped rewards is a significant problem, not just in the application of DDPG (e.g. as in [40]), but for model-free reinforcement learning in general [41]. In this work, we manually design the reward functions used to estimate optimal policies.

Arguably, this is not the most efficient way to find an optimal policy and in fact, several methods exist for combining model-free reinforcement learning with inverse reinforcement learning (IRL) algorithms, which are used to infer a reward function given state-action pairs sampled from an optimal policy [42, 43, 44, 45]. And indeed, this is a interesting direction that we explore throughout the rest of this thesis. We attempted with this work to find a compromise between automating the discovery of good policies and allowing for interpretability of the actions of the learner. IRL presents a powerful set of tools with which policies can be learned to perform complex objectives, but for the first application of deep reinforcement learning to biological systems, we felt it best to ensure that the problems being solved had clear, interpretable control objectives.

Ideally, we would be able to take an optimal policy and deduce either an analytical form or empirical results explaining its performance. In fact, developing methods to allow deep neural networks to "explain themselves" is an active area of research in the machine learning and statistics communities. These methods are being proposed in response to the classical notion put forth famously by Richard Feynman who said, "What I cannot create, I do not understand". This philosophy would suggest to replace mysterious, poorly understood parts of deep reinforcement learning with manually constructed models (e.g. in a similar vein as what is done to improve the sample-complexity of model-free methods by incorporating manually designed components [32]). Alternatively, attempts have been made to leave a black-box machine learning algorithm intact and attempt to better understand it. For example, a method using influence functions was recently developed to yield insight into how deep neural networks work when used for supervised learning [46]; extending methods like that in [46] for problems in reinforcement learning is another interesting direction for future work. Further extensions to alternating IRL/policy optimization solvers (e.g. as in [45]) is an even longer term goal.

There are other factors to consider in the application of deep reinforcement learning algorithms to biological systems in addition to the ability of humans to understand them. For example, our work assumes full observability of the system state. Relaxing this assumption to partial observability is required in some applications (requiring the use of methods such as [47]): the manner in which the uncertainty induced by partial observability interacts with modeling uncertainty is an important problem for applications to biology. The off-policy nature of DDPG allows for a reduced number of samples from the policy during learning and thus, the use of experience replay during learning. This can reduce the number of times a controller would need to interact with an actual brain in order to fit a policy. Another approach for improving the efficiency of exploration was

proposed in [40], where the authors show that data generation and resampling efficiency can be improved relative to the number of parameter updates. For complex objectives, it may be helpful to initialize the search for an optimal policy from states that have achieved a partial reward. Both methods help to accelerate discovery of optimal policies for complex control objectives.

2.4 Experimental Details

2.4.1 Deep Neural Network Parameters

For the approximation of the Q-function as well as the policy and target networks, deep networks with two hidden layers were used. For the network of SLIF neurons, the first hidden layer had 400 units and the second had 300 units, while for the KM, the first hidden layer had 1200 units and the second had 1000. A value of $\tau = 0.001$ was used in the exponential moving average between updated network parameters and past network parameters. ADAM [48] was used for stochastic gradient descent to perform parameter updates. The same learning rate was used for fitting the actor and critic networks for both model systems. For control of the network of SLIF neurons, a learning rate of 0.01 was used, while for control of the KM, a learning rate of 0.0001 was used. These learning rates were chosen based on the rates used in the original DDPG paper (KM control) or slightly modifying these values (SLIF control). The learning rate was increased for SLIF control because it was observed that if this rate were too low, the optimal stimulation strength was never reached and reliable spiking behavior wasn't induced in any of the cells. Mini-batch stochastic gradient descent was performed with a batch size of 32 4-tuples, where each 4-tuple contained $(s_t, a_t, r(s_t, a_t), s_{t+1})$, for some time t. The discounted rate of future returns used was $\gamma = 0.99$.

2.4.2 SLIF Parameters

We used $C = 10.0$, $\tau_v = 15.0$, a resting membrane potential of -70.0, a depolarization threshold of -50.0, a refractory period of length 0., $\tau_s = 1.0$, $\Delta t = 1.0$, $\bar{g} = 0.01$, $\eta = \sqrt{2}$, and $E_{syn} = 70.0$ [9, 10]. All simulations were begun with cells at the resting membrane potential. Before the controller delivered input into the system in both training and testing instances, the network was reinitialized to resting membrane potential. For the fully-actuated example, network adjacency matrices were initialized with directed edges whose weights were drawn from a uniform distribution over $[0, 1]$. This matrix was made symmetric for the under-actuated example. Simulations were run on networks as large as 40 cells for the fully-actuated example; 4 neuron networks were used for the 1-D under-actuated example and 32 neuron networks were used for the 2-D under-actuated example. For the fully-actuated example, the state consisted of a 10 timestep spike history concatenated with a 10 timestep lookahead into the target spike train. Similarly, for the under-actuated example, a 10 timestep spike history was concatenated with a 10 timestep lookahead to construct the state. Since the projection of the binary vector of all neuron spikes onto a principal component effectively gives an instantaneous firing rate of all neurons in a given community, the lookahead in this case consisted of sampled spiking activity of the controlled cell at the target rate.

2.4.3 KM Parameters

The controller was tested on a network of 20 oscillators, each initialized to a phase (ϕ_i) drawn uniformly at random from $[0, 2\pi]$, with natural frequencies (ω_i) drawn from $N(0, 10)$, and edge weights ($A_{i,j}$) drawn from a uniform distribution over $[0, 1]$. For our experiments, we chose $K = 0.1$. ϵ, η were 0.1 and 1.0 respectively.

Chapter 3

Minimum Free Energy Optimization and Motor Learning

Motor learning in biological systems is defined as a change in the capacity to behave, based on experience and practice. The change in behavioral capacity is typically described in terms of improved performance. However, it has become increasingly apparent that an additional important property of movement is persistent performance variability despite extensive training. Indeed, there exists an extensive number of motor control studies of birdsong, locomotion, and limb control demonstrating the extent to which movement variability influences and is influenced by performance and learning [49, 50, 51, 52, 53]. An important theme throughout this work is that even in learned, highly stereo-typed behaviors, there exists variability in the expression of these behaviors both within and across subjects. The presence of systematic variability in behaviors that have been heavily trained poses a problem for understanding how these systems learn. Even in the case where the behavior is generated by a stochastic system, the learning objective cannot simply be a single performance variable such as error minimization (accuracy) or maximal speed: this would result in behavior with zero variability. This suggests that a

more general framework for characterizing learning objectives is necessary to explain a putative aspect of motor learning.

To address this need, we present an approach that tracks dynamic changes of performance (in our study, movement time) while also capturing performance variability in terms of a free energy functional of density dynamics. At the same time, we characterize the evolving dynamics of neural activity, whose variability is also described as a property of a density function. Neural activity is based on fMRI BOLD measurements recorded as subjects learn a set of finger sequences practiced at different training intensities. The goal of this work is to determine how the joint brain-behavior densities evolve as a function of the amount of training.

We show that the dynamics of the density over global (all brain regions) as well as localized (the task-active regions) brain-behavior pairs follow a Fokker-Planck Partial Differential Equation (FPE)(the term density is used as short-hand for the probability density function in this section). The FPE is a fundamental aspect of the physical sciences, both for classical and quantum mechanics [54, 55, 56]. With respect to the neurosciences, the FPE is the population-level version of the Drift-Diffusion equation often used to model decision making [57, 58] and has also been used to model stochastic neuronal dynamics [59]. The advantage of this joint brain-behavior density framework is that it offers a potential explanation of the nature of behavioral variability and how it is tuned during learning. A strength of this explanation is that it is grounded in the dynamics of the underlying neural activity. To the best of our knowledge, the combined modeling of neural activity and behavior is a novel extension of past work on motor variability [49, 50, 51].

The introduction of this joint brain-behavior framework provides a precise formulation

of the learning objective that gives rise to the observed variability. Specifically, we show that the optimization of a popular objective in the Reinforcement Learning and Optimal Control literature [60] also yields dynamics that follow the FPE. This objective is so named because it contains two terms: expected performance of the brain-behavior density and its entropy. We refer to this framework as Minimum Free Energy Learning (MFEL). The consequences of this finding are twofold: first, it suggests an appropriate definition of behavioral variability as the entropy of the brain-behavior density. Next, it suggests a way to recover the parameters of the MFEL objective to infer the performance objective optimized as well as the manner in which variability is tuned during learning.

Using a novel variant of inverse reinforcement learning [1], we retrieve the cost function optimized during motor learning, as well as the parameter tuning the entropy of the brain-behavior density (see Figure 3.1). This allows us to relate the population-level analysis performed to infer these objects to learning on the individual level. In particular, we show that the MFEL framework is appropriate to characterize individual learning by showing that individuals optimize the same objective as the population of subjects.

[1]The appeal to reinforcement learning is meant to highlight the connection between our method and the control of neural systems. In fact, the method presented in the Supplement is a generic approach to parameter estimation for density dynamics following the Fokker-Planck equation.

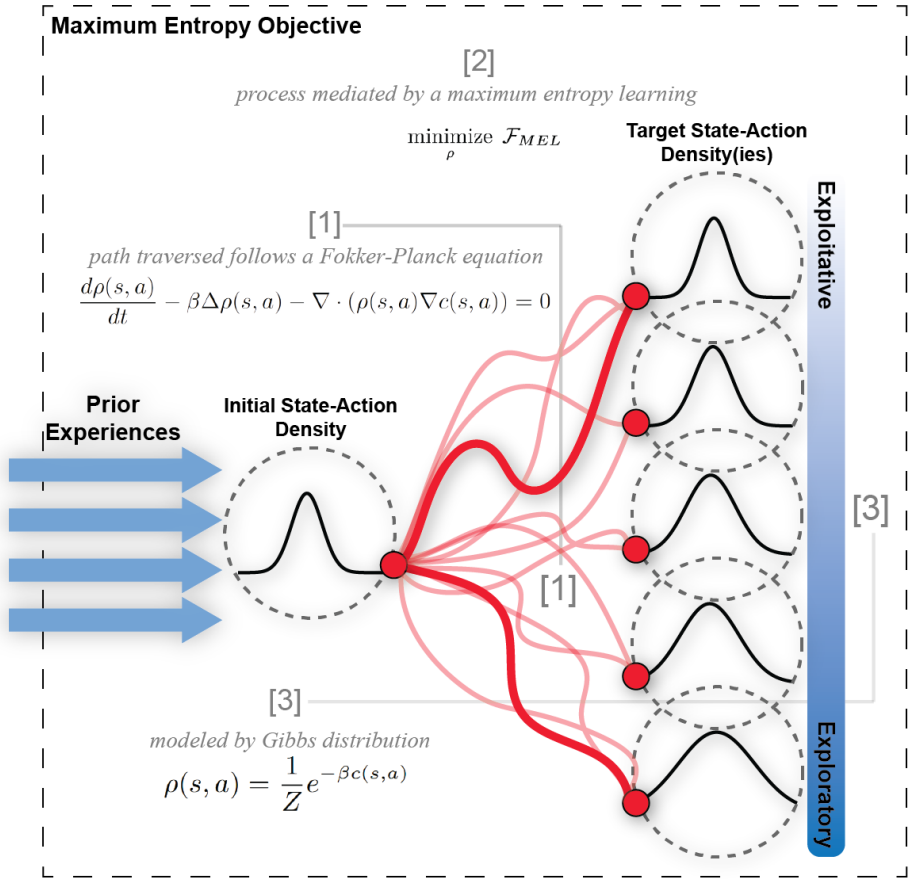


Figure 3.1: Overview of the findings of this work. Previous experience is embodied in the initial brain-behavior density before any learning. Based on the form of the dynamics of this density (red lines, [1]), this density is modeled as a Gibb's distribution ([3]). The eventual target of these dynamics is largely influenced by the temperature parameter, β. This coefficient tunes the entropy of the brain-behavior density, as shown by the Minimum Free Energy Objective (3.3).

3.1 Data Collection

3.1.1 Experimental Design

The motor sequence training protocol occurred over a 6-week period with 4 MRI scanning sessions spaced 2 weeks apart on Day 1, Day 14, Day 28, and Day 42 (Figure 3.2). On Day 1 of the experimental protocol, the participants completed their first MRI session,

Scan 1, and the experimenter installed the training module on the participant's personal laptop and taught them how to use it for at-home training sessions. Behavioral measurements were taken during these at-home training sessions and interspersed throughout this training regimen, neuronal measurements were taken using fMRI BOLD. Participants were required to do the training for a minimum of 10 out of the 14 days in each 2-week period between the subsequent scanning sessions. All participants completed the full training regimen; none completed less than 10 full training sessions.

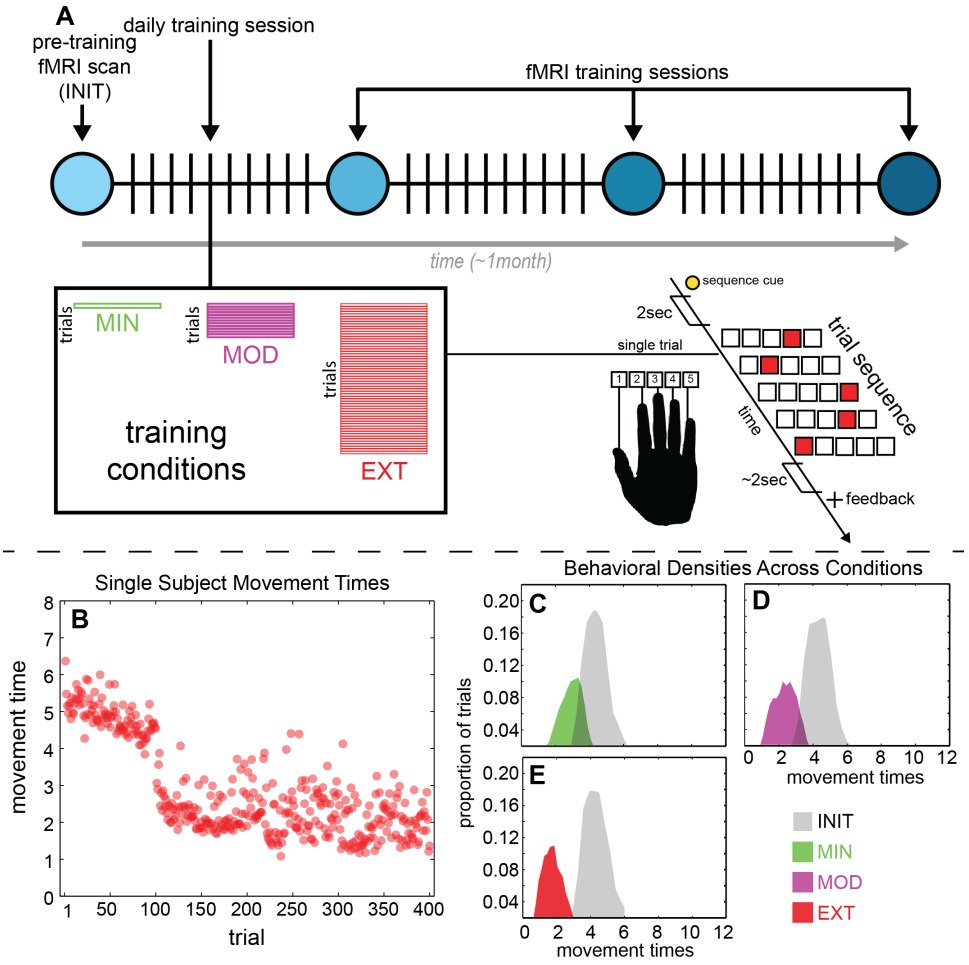


Figure 3.2: **A**: Schematic summarizing the DSP task as well as the experimental design. **B**: Example movement times for a single subject in the EXT condition. Each scanning session consists of 100 trials, and each point gives the performance on a single trial. Movement time variability persists even with the highest training intensity and duration. **C,D,E**: Evolution of movement time densities for EXT (E), MOD (D), and MIN (C) conditions with INIT shown in gray.

In their at-home training sessions, participants practiced a set of 10-element sequences using their right hand. Sequences were presented using a horizontal array of 5 square stimuli, and the key responses were mapped from left to right, such that the thumb corresponded to the leftmost stimulus and the pinky finger corresponded to the rightmost stimulus (Figure 3.2). A square highlighted in red served as the target stimulus, and the next square in the sequence was highlighted immediately after each correct key press. If

an incorrect key was pressed, the sequence was paused at the error and restarted upon the appropriate key press. Participants had an unlimited amount of time to respond and complete each trial.

Each practice trial began with the presentation of a sequence-identity cue that identified 1 of 6 sequences. These 6 sequences were presented with 3 different levels of exposure, in order to acquire data over a larger range of learning stages while controlling for the effect of scanning day. The 2 extensively trained (EXT) sequences were identified with a colored circle (cyan for sequence A and magenta for B), and they were each practiced for 64 trials during every at-home training session. The 2 moderately trained (MOD) sequences were identified by triangles (red for sequence C and green for D) and each practiced for 10 trials in every session. The 2 minimally trained (MIN) sequences were identified by black outlined stars (filled with orange for sequence E and white for F) and only practiced for 1 trial each during the at-home training sessions. Participants were given feedback every 10 trials that reported the number of error-free sequences and the mean time required to complete them.

3.1.2 Data Collection

Twenty-two right-handed participants (13 females and 9 males; mean age, 24 years) volunteered and provided informed consent in writing in accordance with the guidelines of the Institutional Review Board of the University of California, Santa Barbara. All had normal or corrected vision and no history of neurological disease or psychiatric disorders. We excluded 2 participants because 1 participant failed to complete the experiment and the other exhibited excessive head motion (persistent head motion greater than 5 mm during the MRI scanning).

During each of the 4 MRI scanning sessions, we collected functional echo planar imaging data while participants performed 300 trials of the self-paced motor sequence task. Unlike the at-home practice sessions, participants completed an equal number of trials for each of the three exposure types. The 50 trials for each sequence type were grouped in blocks of 10 trials of the same sequence type (10 MIN, 10 MOD, 10 EXT), and the blocks were randomly ordered across the 5 BOLD runs. After each block of 10, participants received feedback about the number of error-free sequences and mean reaction time to complete the sequences.

Because sequence production was self-paced, the number of scanned TRs varied between subject and session. In order to collect event-related fMRI data, the inter-trial interval ranged between 0 and 6 s (average of 5 s). The number of sequence trials performed during each scan session was the same for all subjects with the exception of 2 abbreviated sessions due to technical problems. In each of these 2 cases, the scanning protocol was stopped short, so that 4 out of the normally acquired 5 runs were completed. Data from these sessions are included in the presented analysis.

3.1.3 fMRI Data Analysis

Functional imaging data processing and analysis was performed using Statistical Parametric Mapping (SPM8, Wellcome Department of Cognitive Neurology). Raw functional data were realigned, co-registered to the native T1 (using the first mean image as the base image for all functional scans), normalized to the MNI-152 template with a re-sliced resolution of $3 \times 3 \times 3$ mm, and then smoothed with a kernel of 8 mm full-width at half-maximum.

BOLD response was modeled for each subject using a single design matrix with parameters estimated using a general linear model (GLM). An event-related design was used to model sequence-specific activity patterns. Trial onset is signaled by the presentation of the sequence identity cue and is presented 2 s prior to the initial DSP target stimulus. Neural activity in this case reflects both the preparation and production of learned sequences. The design matrix for each subject was constructed using separate factors for each scan session (pre-training, training sessions 1-3), exposure condition (MIN, MOD, and EXT), and repetition (new or repeated trial). A trial is coded as a repeated event if the previous trial was the exact same sequence and the previous trial had been performed correctly. Error trials and repeated trials that followed error trials were modeled using a separate column in the design matrix. Blocking variables were used to account for non-specific session effects for each scan run.

The full-factorial design option in SPM was used to perform higher-level mixed-effects group analysis. Skill-specific longitudinal effects were modeled using a single factor (12 levels: one for each exposure condition and session). Training intensity, that is, the cumulative amount of training trials performed were used for model factor levels: pre-training (MIN/MOD/EXT), MIN during training scans 1-3, MOD during training scans 1-3, and EXT during training scans 1-3. We were primarily interested in analyzing BOLD dynamics with respect to training intensity. To do this, a contrast was developed at the group level where the main effect of training intensity, over all sequences, scanning sessions, and types of training intensity, was calculated using a one-sample t-test and corrected for multiple comparisons using family-wise error (FWE) correction ($P < 0.05$).

Based on the previous literature on motor learning, we focused our analysis on 9 senso-

rimotor regions including the postcentral gyrus, supplementary motor area, and lateral occipital cortices [61]. To investigate neural activation within these areas during the task, we constructed a mask image which represented the intersection of each brain region as indicated by the Harvard-Oxford atlas and the group level contrast of training intensity that was FWE-corrected. This ensured that we analyzed the task-active voxels within the sensorimotor regions that were common across the group, and then we extracted an average timeseries for each individual from each region for each training intensity. This provided a matrix that was 24 (subjects) x 9 (sensorimotor regions) x 3 (training intensity MIN, MOD, and EXT).

The estimated beta weights reflect a group-level GLM contrast that reflects the main effect of training intensity across all sequences, scanning, sessions, and types of training intensity. This map was the result from a one-sample t-test corrected for multiple comparisons using a family-wise-error (FWE) correction using a p-value threshold of 0.05. The higher-level group mixed-effects model was estimated using all but one subject, and from this, the identified local optima were used to extract mean beta weights from the remaining subject. Mean beta weights were extracted using a spherical ROI (6 mm radius) centered on each local optimum. We performed this procedure for each of the 20 subjects, so that the displayed amplitudes correspond to the overall mean of the left out subjects' beta weights.

The brain regions outside the sensorimotor system were defined based on task active voxels in the group level contrast reflecting the main effect of training intensity across al sequences, scanning sessions, and types of training intensity. A mask image was constructed which represented the intersection of each Harvard-Oxford anatomical region and the group level task activation image, and then the mask was applied to each subject's

image. For each subject, the mean of the extracted, non-zero voxels within each region for each subject and each condition was computed.

3.2 Behavioral Variability Persists During Motor Learning

First, we examine behavioral performance during a motor learning task as a function of training intensity. Across all participants and all training intensity conditions, there was a reduction in movement time during the motor learning task (see Fig 3.2C-E). A significant decrease in average movement time across all training intensity conditions relative to the initial training session was observed (see Fig 3.2C; MIN vs INIT, M= 1.88, SD= 0.63, t(df)=139.83, p<0.0001; MOD vs INIT, M= 2.49, SD=0.75, t(df)=96.86, p<0.0001); EXT vs INIT, M=3.04, SD=0.71, t(df)=74.65, p<0.0001), with average performance on sequences in the EXT condition showing the greatest reduction in movement times relative to initial training. In fact, alternative hypotheses were rejected using T-tests when all pairs of the four conditions were compared with each other, rather than just comparing MIN, MOD, and EXT conditions with INIT (p-values were less than 0.0001). This demonstrates that the exposure of individuals to more intense training will improve their performance as defined by the average movement time.

In addition to improved average performance, motor learning is characterized by the persistence of behavioral variability. We refer to this variability as the entropy of the behavioral density. This definition is formally justified in Section C. The experimental data (Figure 3.2B) suggests that analyzing the dynamics of the density over movement times, and its entropy in particular, may help to explain the origin of this variability and

allow us to understand its evolution over time.

To this point, the evolution of the movement time densities as a function of training intensity is shown in Figures 3.2C, 3.2D, and 3.2E. Notably, the entropy in the movement time density does not decrease to zero with increased training intensity. We relate this result to past work where even highly trained, stereotyped behaviors retain a certain amount of variability when executed [49, 50, 51, 52, 53]. The fact that the entropy of the movement time density after high training intensity is non-zero suggests that learning has at least two objectives: one is improved average performance and the other is tuning the entropy of the density. This follows from the fact that simply optimizing for average performance would result in deterministic behavior (i.e. a movement time density with zero entropy). This is not to say that behavioral variability is intentionally preserved by the brain, but it may be that there is a minimum amount of noise in the execution of movements that cannot be further refined. Yet, even in this case, in order to accurately model motor learning, this persistent noise must be mathematically formulated and incorporated into the model.

3.3 Motor Learning Follows Fokker-Planck Dynamics

To examine the dynamics of the neural substrates of motor learning across all regions involved in sequence production, BOLD beta values from task-dependent brain regions were extracted using the Harvard-Oxford atlas. In Figure 3.3, the densities of BOLD beta values are plotted to demonstrate the changes in global brain dynamics across the different training intensity conditions. There was a decrease in the entropy of the BOLD

density relative to initial training, but similar to behavioral performance results, this entropy remains non-zero at the highest training intensity (see Figure 3.3B; INIT, M=0.19, SD=2.78; MIN vs INIT, M=0.007, SD=2.16, Levene=6.47, p=0.012; MOD vs INIT, M=0.0168, SD=2.05, Levene=8.50, p=0.004); EXT vs INIT, M=0.086, SD=1.81, Levene=15.14, p=0.0001).

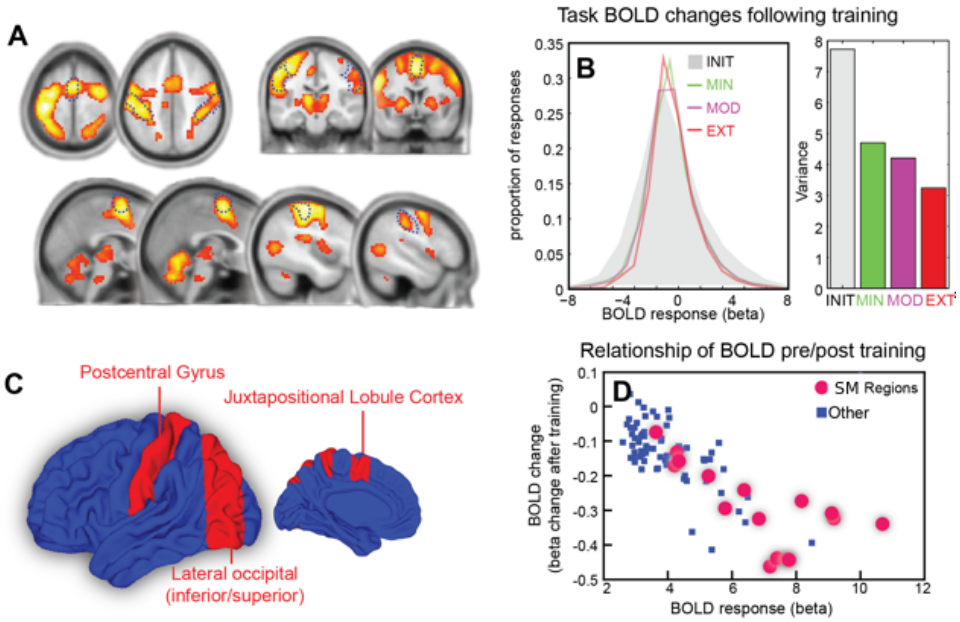


Figure 3.3: **A**: Global task-active beta coefficients illustrated on axial, sagittal, and coronal slices (FWE < 0.05). **B**: Global density of beta coefficients (left) and the variances of the densities for each training intensity. **C**: Sensorimotor (SM) brain regions (red) and all other brain regions (blue). **D**: Change in beta coefficient with training intensity plotted against initial beta coefficient for SM regions (red) and all other task-active regions (blue).

Importantly, brain regions that are implicated in sensorimotor function are more sensitive to these dynamics than other task-relevant brain areas. This is shown in 3.3D, where the change in the beta coefficient with increased training intensity is plotted against the initial beta coefficient, and a 2-Dimensional Kolmogorov-Smirnov Test distinguishes these two groups of brain regions with p-value of 0.00031. This result also holds at the indi-

vidual level for all but four subjects (with p-values of 0.05926, 0.11123, 0.18631, 0.11049 respectively).

Naively, this decreasing entropy seen at the global scale might be explained by a minimization of extraneous and error-prone movements and a refinement of movements to more efficiently execute each sequence. But the dynamics of the movement time density in Figures 3.2C, 3.2D, and 3.2E suggest that the influence of training intensity is more subtle. Simply optimizing for performance (movement time) would result in deterministic behavior. The fact that even expert behavior on EXT sequences is probabilistically distributed suggests that a different model of learning is required.

To better visualize the relationship between neural activity and movement time, we plot the brain-behavior density as it evolves with increased training intensity in Figure 3.4. A Partial Differential Equation (PDE) that captures the dynamics shown is the Fokker-Planck equation (FPE), which is given by

$$\frac{d\rho(b)}{dI} - \beta\Delta\rho(b) - \nabla \cdot (\rho(b)\nabla c(b)) = 0, \tag{3.1}$$

where b is the 2-tuple containing the random variables for neural activity and behavior, $\rho(b)$ is the probability density over brain-behavior pairs, Δ is the Laplacian operator, $\nabla\cdot$ is the divergence operator, $c(b)$ is a cost function, β is the diffusion coefficient, and I is the training intensity. This equation can be understood as shifting an initial value of $\rho(b)$ (corresponding to the INIT condition) in the direction specified by $\nabla c(b)$ while producing diffusion, the direction and rate of which is specified by β (we relate the diffusion of $\rho(b)$ to entropy in the next section). We have defined the evolution of the density with respect to training intensity, though the FPE is typically used to characterize the evolution of a

density with respect to time. Since we have also defined training intensity as the number of exposures of a subject to a sequence, assuming each exposure takes a fixed amount of time, these two approaches can be seen as equivalent.

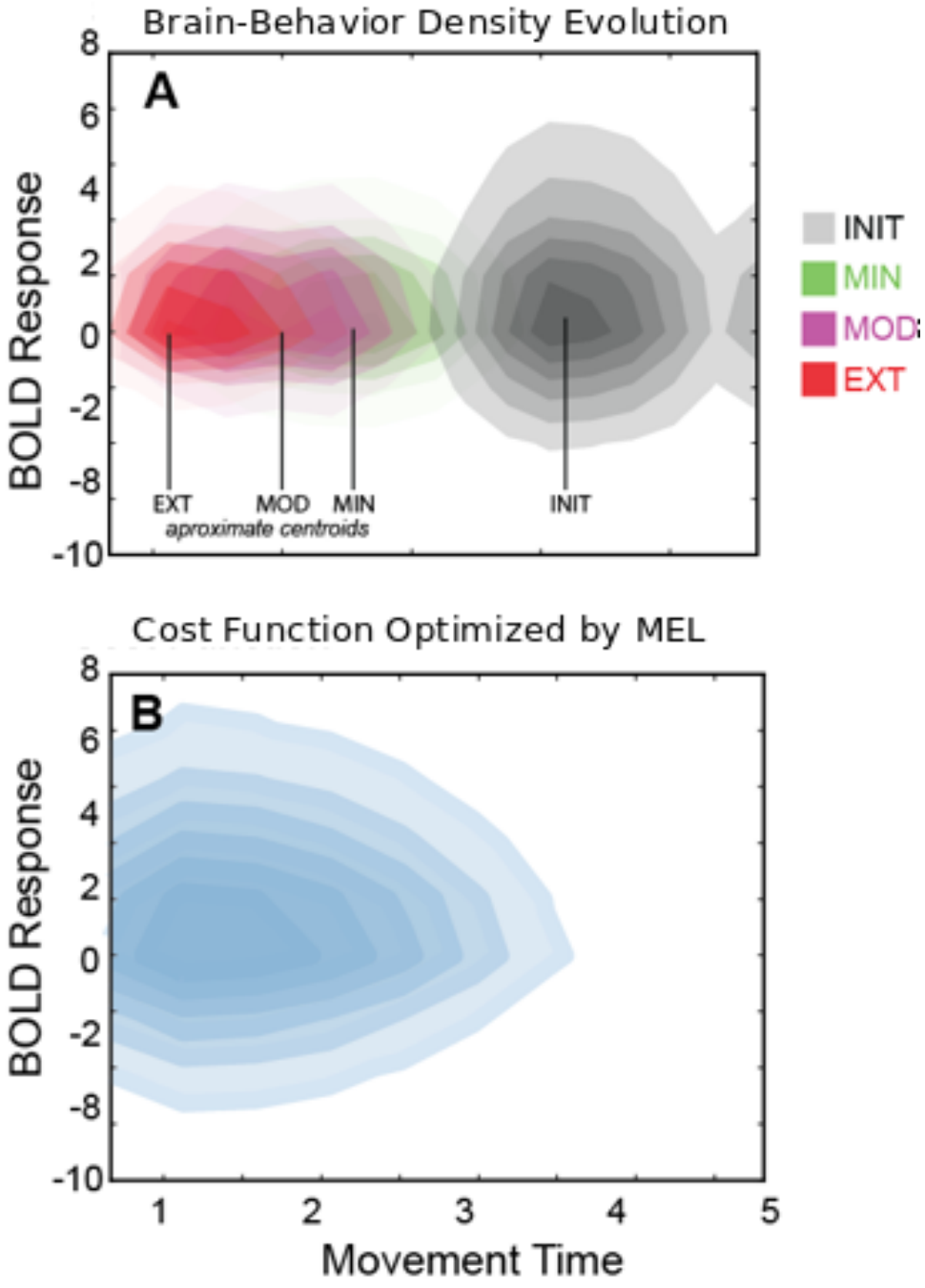


Figure 3.4: **A**: Evolution of the brain-behavior densities with increased training intensity. **B**: The cost function optimized during MFEL as derived from population level analysis.

In the case of the DSP task presented in this work, the cost function is the mathematical

representation of the motivation each subject has to improve his respective performance on the task. Put simply, $\rho(b)$ performs steepest descent on $c(b)$ to improve performance and change the shape of $\rho(b)$, while the diffusion term tunes the entropy of $\rho(b)$. The incorporation of this function into the FPE framework not only gives insight as to the dynamics of the brain-behavior densities (goodness of fit tests are provided in the Supplement), but also the rate at which the brain-behavior densities converge to an expert state during learning. In the Supplement, we show that the solution of the FPE converges to steady-state exponentially fast, explaining the exponential improvement seen in the behavioral performance of the subjects.

Previous work in neuroscience and physics has demonstrated that the use of Fokker-Planck dynamics is a biologically appropriate model for explaining stochastic neuronal dynamics [59]. While a majority of prior work has primarily focused on using the FPE to model stochastic changes in neuronal networks, the present study extends this line of research to explain the neurobehavioral dynamics of motor learning through training. Specifically, we extend the use of the FPE to show that it applies to jointly model the BOLD response and behavior of subjects, a result that doesn't necessarily follow from past work on neuronal dynamics. In the context of motor learning, the FPE provides a mathematical framework to precisely define the source of and mechanism for tuning behavioral variability: both derive from diffusion of the brain-behavior density.

3.4 Fokker-Planck Dynamics are Generated via Free Energy Optimization

While the FPE may capture the dynamics of the brain-behavior density during learning, it is not clear how these dynamics relate to the problem solved by the subjects. In fact, the FPE suggests a popular framework as a model for the learning problem solved by the subjects. To proceed further though, we require a model of the brain-behavior density. The steady-state distribution of the Fokker-Planck equation is the Gibbs distribution

$$\rho(b) = \frac{1}{Z} e^{-\beta c(b)}, \tag{3.2}$$

where Z is a normalizing constant and β is a temperature parameter. This distribution also appears in the literature under another name, the "Maximum Entropy Distribution". There are many ways to interpret this name, but perhaps the most direct is to begin with an optimization problem. Consider the objective

$$\mathcal{F}_{MFEL} = \mathrm{E}_{\rho}[c] - \beta \mathrm{H}[\rho], \tag{3.3}$$

where H is the entropy of ρ, E_{ρ} is the expectation operator with respect to the brain-behavior density ρ, and c is the cost as defined in the previous section. Equation 3.3 is actually a specific example of a more general expression [62, 63]. In particular, c may be redefined as a generic "potential" or "energy", ϕ. In its current form with ϕ interpreted as a cost, this equation is commonly used for policy optimization methods in reinforcement learning and control engineering [64, 65, 66] and can be related to models of neural systems as optimizing prediction errors [67]. By setting ϕ to be the negative log-likelihood of the data, Equation 3.3 can also be used to derive the Evidence Lower Bound (ELBO), used for variational inference. This objective can be incorporated into

the following optimization problem

$$\underset{\rho}{\text{minimize}}\ \mathcal{F}_{MFEL}. \tag{3.4}$$

If a brain-behavior density is found by optimizing this expression in a particular way, then its dynamics follow the Fokker-Planck equation (see Supplement for derivation). Because the dynamics of the brain-behavior density follow a Fokker-Planck equation during motor learning, Equation 3.4 is also a good model of the optimization problem that accounts for the neural changes during learning. This connection formalizes the intuition given in the previous section: motor learning proceeds via simultaneous optimization of expected cost and the brain-behavior density. We refer to this model as *Minimum Free Energy Learning* (MFEL) throughout the rest of this chapter.

The MFEL model implies that behavioral variability is tuned by adjusting the entropy of the brain-behavior density (i.e. tuning β). For example, if the entropy of the brain-behavior density is increased along the behavioral coordinate, then samples from this density are going to be more variable. Given empirical examples of the evolution of the brain-behavior density though, it is not immediately clear how to estimate the cost function, $c(b)$, or the temperature parameter, β. These objects are retrieved in the next section (the methods for doing so are presented in the Supplement) and the use of the MFEL framework as a model for motor learning is further validated.

3.5 Each Subject Learns the Same Optimal Behavior

The objective given in equation 3.4 represents a rule governing how the population of subjects learns. But when analyzing the learning procedure of individual subjects, both

the structure of the brain-behavior density and its dynamics might seem quite different from those presented in Figure 3.4. In order to validate the utility of the population-level analysis for modeling learning within individual subjects, we first inferred the structure of the cost function optimized during motor learning. To do this, we developed a novel approach to Inverse Reinforcement Learning (IRL) in order to compute an explicit representation of the cost function. One class of IRL methods, called Maximum Entropy IRL (MEIRL), attempt to infer $c(b)$ given samples from $p^*(b)$, assuming that $p^*(b)$ has the form of the Gibb's distribution. One strategy for finding $c(b)$ in this case is to use a gradient-based method to optimize the negative log-likelihood of the samples (e.g. [45]). This approach is not ideal in the case of the data presented here because the optimization scheme does not necessarily preserve the Fokker-Planck dynamics observed during learning. Instead, we would like to develop a method which not only retrieves $c(b)$, but does so in a way that is consistent with the dynamics of neural learning.

The method we develop relies on the modification of a popular method used to simulate the FPE. Briefly, this method simulates the FPE by solving a sequence of optimal transport problems. That is, one can simulate the FPE by, at every timestep t, evolving $p_t(b)$ to $p_{t+1}(b)$ by finding the $p_{t+1}(b)$ that is as close as possible to $p_t(b)$, while still reducing the value of equation 3.3. Since, for our data, $p_{t+1}(b)$ is known (i.e. it is either the empirical densities for the MIN, MOD, or EXT conditions), we simply need to solve the optimal transport problem between $p_t(b)$ and $p_{t+1}(b)$. The cost function optimized in moving from $p_t(b)$ to $p_{t+1}(b)$ can be retrieved from the solutiom of these optimal transport problems (see Supplement for full derivation).

The cost function returned by our method is shown in Figure 3.4. The cost function is approximately convex, and this result implies, given the MFEL model, that the opti-

mal brain-behavior density is always achieved and this density is unique. With respect to tuning behavioral variability, this theoretical guarantee indicates that there is an optimal level of variability (i.e. an optimal value for the entropy of the brain-behavior density). This follows from the fact that this cost function includes β from equation 3.2. So, in effect, it includes information on the cost function being optimized as well as the target variability.

Finally, we present evidence demonstrating that each subject optimizes a similar objective in Figure 3.5. Using the cost function derived from the population-level analysis (i.e. shown in the bottom plot of Figure 3.4) in the objective in equation 3.4, we computed this objective using the brain-behavior densities for each individual subject. The curves presented in Figure 3.5 demonstrate that there is exponential improvement in this objective with increased training intensity. Moreover, we note that nearly every single subject demonstrated strictly monotonic improvement in this objective with increased training intensity. These results suggest that the estimate of the objective given in equation 3.4 is not only a good representation of population-level learning, but also of learning that takes place within the individual.

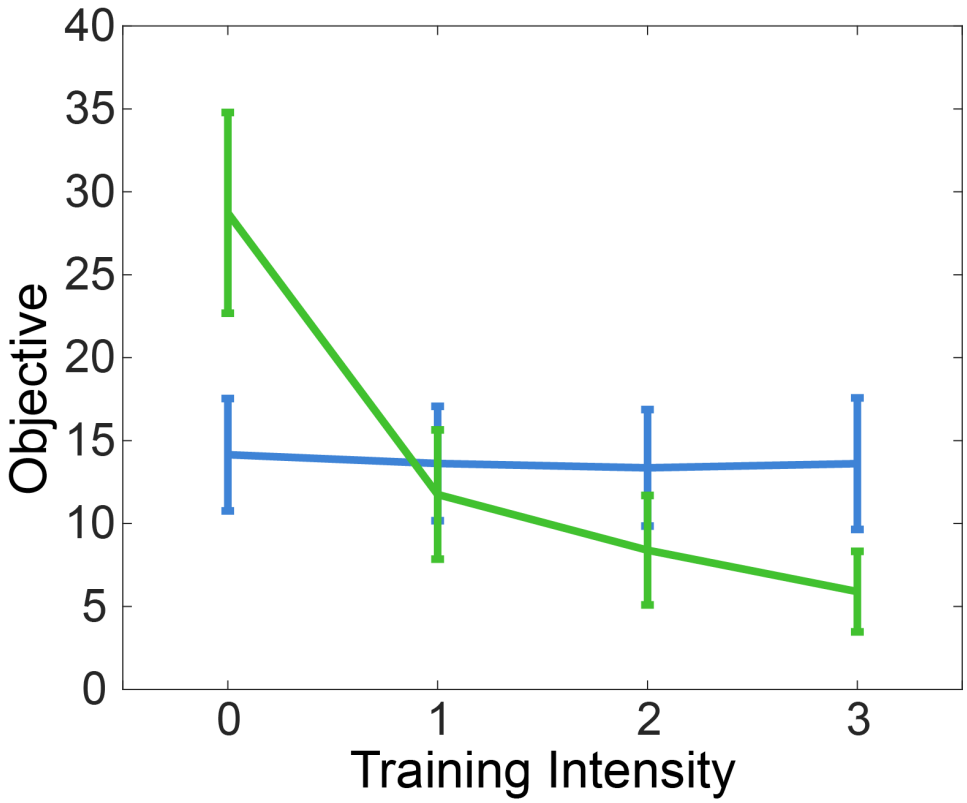


Figure 3.5: The value of the population-level objective for each subject plotted against training intensity (**green**) compared with a null model (**blue**). The null model was generated by shuffling data across conditions, within each subjects dataset. The brain-behavior density for each subject is evaluated using the MFEL objective, where the cost function is derived from a population-level analysis. This plot demonstrates that individual subjects optimize the population-level objective.

3.6 Discussion

In this work, we presented a Minimum Free Energy model of motor learning. We have taken a popular objective from the engineering and statistics community, Equation 3.4, and shown that it has a strong biological foundation. This connection between biology and engineering follows from the evolution of the empirical brain-behavior density according to a Fokker-Planck equation. We show how learning is characterized as evolving with training intensity and analyze the full density of observed responses (behavioral

and neuronal) to justify this perspective. In doing so, we are able to connect a number of seemingly disparate schools of thought: the brain as a controller, inference engine, and dynamical system [68, 69, 70]. Optimization of Equation 3.4 is commonly used to perform policy optimization for control systems. This problem is equivalent to the one solved during Variational Bayesian Inference, where the cost (c) is interpreted as a negative log-likelihood and a KL-Divergence is minimized between estimated and actual posterior densities. And we show that if Equation 3.4 is solved in a particular manner, the dynamics follow the FPE. The ability of the MFEL to act as a unifying principle across these three schools of thought allows us to lend support to the theory of learning in the brain as performing Bayesian inference. The brain-behavior densities for a given training intensity can be interpreted as posterior densities where responses are sampled from these densities. The fact that this kind of inference is performed by optimizing the entropy of Bayesian beliefs about responses speaks to the close connection between the Minimum Free Energy and Occam's principles.

The fundamental finding that motor learning can be modeled with an FPE explains why it proceeds exponentially fast, as well as suggests a new approach towards solving the IRL problem. Many different approaches towards IRL have been taken, but it is not clear how any of them relate to the dynamics of learning. This raises issues related to the accuracy of the cost function retrieved, especially in the case where the solution is not unique. Our approach, based on optimal transport, is both novel and has a clear connection with the biology of learning. While optimal transport has been used for Bayesian inference, it hasn't been used for IRL [71]. More importantly, we are able to use the population-level inferences and show that individual subjects also optimize the population-level objective. We thus are able to give both theoretical and empirical support for the use of our variant of IRL.

Another conclusion following from the observed learning dynamics is that samples from the brain-behavior density become less variable with increased training intensity. This can be interpreted as a kind of inflexibility, a concept which has been studied from a number of different perspectives previously, including a network scientific perspective [72, 73, 74, 75], aging [76], and neural systems at the cellular and circuit levels [77]. With respect to artificial controllers fit using the MFEL objective, this inflexibility manifests as an inability of learned policies to handle non-stationary environments (i.e. a cost function that changes with time) [29]. It is not clear though that inflexibility in organic neural controllers fit using the MFEL objective would be a direct consequence of exposing subjects to increased training intensity.

One complication that might arise involves the ability of organic neural systems to maintain multiple skills at once, though it is currently an open problem to train artificial neural network controllers to do the same [78]. Inflexibility is not necessarily problematic in the case of a stationary environment. Inflexibility must thus take into account the size of the space of skills a neural controller has learned and the probability of the environment to transition away from this space. Further work is required to combine these ideas with the models of MFEL presented here.

The study of individual differences with respect to the MFEL model presented in this chapter is another interesting avenue for future research. We showed in Figure 3.5 that individual subjects largely demonstrate exponential improvement in the population-level objective with increased training intensity. Within this population of subjects though, there is a non-zero variance over the learning rate (i.e. the value λ, if the value of the objective over intensity decays like $e^{-\lambda I}$). From the models presented in this chapter, it is

not immediately clear how to relate MFEL models of individual learning to MFEL models at the population level. A Bayesian approach may be fruitful, where brain-behavior densities of individual subjects are assumed to belong to a family of densities with a common prior. In this case, one would expect to be able to analyze the population-level cost function to provide insight into learning of the entire task and learning dynamics in general, as has been done in this work. Further insight could be drawn based on the structure of individual cost functions though. For example, such insight would include a better understanding of why some individuals tend to have more variable behaviors than others.

Chapter 4

Distributional Temporal Difference Models

The human brain is capable of controlling movement to achieve adaptation to a changing environment extremely quickly. This adaptation is much faster and more flexible than controllers engineered by humans can achieve, in part because our understanding of how human motor control works is incomplete. We argue that methods that can be applied to both biological and artificial systems are necessary in order to bridge this gap [79]. In particular, one of the gaps is the lack of models relating the behavioral errors optimized during adaptation with neural activity. Specifically, even after a large amount of training, behavior is stochastic and the variability of this behavior has been correlated with performance [49, 50, 51, 52, 53]. Because of the persistent variability of behavior, feedback rewards and errors, which are functions of behavior, are probabilistically distributed. Motor learning would then best be framed in terms of the optimization of a *distribution* of rewards or errors. And yet, to the best of our knowledge, there are no known approaches for modeling distributions of rewards during motor learning and relating their optimization to neural activity.

One common approach is to reduce the probabilistic nature of observed rewards to a deterministic function by looking at the expected reward [80, 81, 82, 83, 84]. We argue in this work that such approaches are not a complete representation of the learning process. For example, some neural systems have been shown to optimize the expected future reward, but others may have different, independent objectives: these may include variance reduction or risk-averse learning which involves optimizing the size of the tail(s) of the distribution over future rewards. Moreover, there is a growing body of work that shows that the brain optimizes a Minimum Free Energy (MFE) objective during learning [67, 66, 70, 69, 68, 64, 63, 62, 65, 60, 85]. This objective is equivalent to optimizing the KL-Divergence between error and ideal probability distributions. We contribute results to the body of work on MFE theory by modeling rewards as random variables and proposing that the brain is adapting by minimizing deviations between error and ideal distributions of rewards.

We present behavioral and fMRI BOLD data generated from analysis of 16 subjects, each instructed to minimize the rotation of an unbalanced object at and following its lift. At regular intervals, the center of mass of this object is rotated 180 degrees along its vertical plane, forcing the subjects to adapt their strategy to lift the object while minimizing its roll. In this work, we study adaptation to a changing environment over a series of trials (i.e. a series of attempted lifts). We define the state of the system to be the maximum magnitude of the roll of the object for each trial. Our goal is to model the learning objective that is driving the system to the zero state (the target state). A recent extension of Temporal Difference Learning, called the Temporal Difference Model (TDM) framework, suggests a way to incorporate "closeness" between the current and a target state into a value function [86]. Specifically, if the negative distance from the current to the target state is used as the reward, then the value function quantifies the

expected future proximity to the target state. Stochasticity of the reward function can be modeled using Distributional Reinforcement Learning (DRL), where the reward is modeled as a random variable. We combine the DRL and TDM approaches in this work and refer to the complete model as a Distributional Temporal Difference Model (DTDM). DTDM requires the estimation of a value distribution, rather than a value function, which intuitively corresponds to the distribution over future distances from the target. To fit this distribution, temporal differences between an updated value distribution and a past estimate of the value distribution are used; this is in contrast to classic Temporal Difference Learning which relies on comparisons between value functions. Distributional temporal differences can be interpreted as error signals and we show in this work that the optimization of these errors serves as a good model of motor adaptation.

We treat our experimental set up as a short-time horizon problem where the value distribution models distances between the current and target states at the next trial. We show that the value distribution becomes significantly distorted after a change in the center of mass of the object, a distortion which is quickly corrected after a few trials. This correction involves a shift in the mean of the value distribution, in addition to other changes in the structure of the distribution, including a reduction in variance and a shrinking of the size of its tails. To find a neural basis for all of these different characteristics and potential objectives, we look at the *global* neural activity. We show that the magnitude of the distortion of the value distribution varies continuously with the average deviation in global neural activity, suggesting that the brain is optimizing the distortion in the value distribution during motor adaptation. Further, we show that deviations in global neural activity are directly proportional to those of sensorimotor activity, justifying our choice of representation.

We further validate the utility of the distributional temporal difference by using it to train a robot to perform a similar task, that is, to lift an object with minimal roll and do so while adapting to changes in its center of mass. We use the DTDM to update a model of system dynamics for use in Model Predictive Control (MPC), and as seen in human subjects, our optimization scheme results in exponential improvement of the model, both during initial training and during updating. We show that with this prediction error, the robot is able to quickly update its model and minimize the roll of the object.

4.1 Data Collection

4.1.1 Summary

In our study of motor adaptation, participants (N=16) performed an object lifting task during fMRI scans that required them to minimize the rotation of the object at and during lift. Subjects had to adapt their strategy to the changing of the object's center of mass at regular intervals. Participants performed 7 runs of 40 trials, where each trial required them to use their thumb and index finger to vertically lift an inverted T-shaped object with an unbalanced center of mass while minimizing its roll at lift onset. Each trial required the subjects to lift the object 5 cm from a flat surface and subjects were notified when the magnitude of the roll of the object exceeded 5 Every 10 trials, the object was rotated by 180 degrees, requiring the subjects to change their digit positioning, digit load force, or some combination of the two to achieve task success. For each of the 7 unconstrained runs, subjects were free to change the positioning of their thumb and index finger at will. The position of the thumb, index finger and object (and its roll) was tracked during the course of each trial using a 3-camera motion tracking system. Performance was measured by the absolute maximum magnitude roll generated within

250ms following lift onset (when the object was lifted 1mm from the table). To allow the subjects to familiarize themselves with the experiment, the first run of 40 trials was allocated for practice and no BOLD activity was measured. For the final 6 runs, BOLD activity was collected for all subjects during all trials. Whole-brain analysis was conducted to identify brain regions activated during 17 time bins, each being 400 ms long, beginning 1.2 s before lift onset. For each block of 20 trials, blocks of contiguous trials were averaged to yield 7 conditions: pre-rotation conditions containing trials 2-4, 5-7, and 8-10; a rotation condition containing trial 11; and post-rotation conditions containing trials 12-14, 15-17, and 18-20. This was done to smooth over short-time variation between trials. Beta values from whole brain analysis were extracted using the Juelich atlas. The vector of all beta values is what we refer to as "global neural activity" in this work.

4.1.2 Participants

Twenty healthy subjects participated in this study (median age: 22 years; range: 18 – 32; 11 women). They were right-handed and had normal or correct to normal vision. We excluded four subjects as a result of equipment failure (n = 3) and not finishing the experiment (n = 1). Subjects gave written informed consent and all study procedures were approved by the Human Subjects Committee, Office of Research, University of California–Santa Barbara.

4.1.3 Materials, Design, and Procedure

Subjects were in supine position in the scanner. Excessive head and body motion was minimized with firm cushion padding of the head, neck, and shoulders. Sandbags under the upper right arm minimized upper limb movement. T1 and T2*-weighted scans were

collected followed by BOLD measurements while subjects manipulated a symmetrically-shaped object with a hidden asymmetric mass distribution with the aim of preventing object roll.

Specifications of the custom-made inverted T-shaped object with constrained and unconstrained grasp surfaces along its vertical axis can be found in [87]. In short, the object had a horizontal base and a vertical Plexiglass column. On either side of the vertical column were grip surfaces that were either circular (for constrained contact points) or rectangular (for unconstrained contact points) in shape. A brass block, concealed by covers, was positioned on the horizontal base on either side of the vertical column, creating an asymmetric mass distribution (object torque = 180 Newton millimeter (Nmm)). The total mass of the object was 610g.

The object was placed at arm's length on a table that was placed over the hips of the subject. The object start position was rotated in a counterclockwise direction at a 30° offset from the edge of the table. This position minimized biomechanical constraints that influence object roll (the wrist would be stiffened more when picking up the object when facing forward rather than angled; the former would minimize the object rolling in a clockwise direction). Subjects were asked to press a button that was in a fixed position toward the right of the object between trials. A mirror attached to the head coil gave continuous viewing of the object and the subject's hand.

Anatomical and fMRI data were collected using a Siemens 3T Magnetom Prisma Fit (64-channel phased-array head coil). High-resolution 0.94 mm isotropic T1-weighted (TR = 2500ms, TE = 2.22 ms, FA = 7, FOV = 241 mm) and T2*-weighted (TR = 3200 ms, TE = 566 ms, FOV = 241 mm) whole-brain sagital sequence images were taken. Dur-

ing object manipulation, BOLD contrast was measured with a multi-band T2*-weighted echoplanar gradient-echo imaging sequence (TR = 400 ms, TE = 35 ms, FA = 52, FOV = 192 mm, multi-band factor 8). A functional image contained 48 slices acquired parallel to the AC–PC plane (3 mm thick; 3 × 3 mm in-plane resolution).

The position and roll of the object were measured using three motion tracking cameras that were radiofrequency-shielded (Precision Point Tracking System, Worldviz; see [87] for the in-scanner setup). With this system, we recorded positions with six degrees of freedom using near-infrared LEDs (frame rate: 150 Hz; camera resolution: 640 × 480 VGA; at the focal distance, the spatial accuracy is sub-millimeter). An individual LED marker was positioned on either side of the T-shaped object on the outer tip of the aluminum rods (to measure object roll).

4.1.4 Experimental Design and Procedure

The experimental task consisted of four conditions: manipulating the left- and right-weighted object at constrained and unconstrained contact points. Before scanning, subjects completed 40 practice trials to familiarize them with the audio cues instructing when and how to lift the object on a given trial. The 40 trials consisted of 10 blocked trials for each of the 4 conditions (20 trials at unconstrained and 20 trials at constrained grasp contact points). We focus on the data generated from the unconstrained trials in this work.

Each trial began with the subject's hand relaxed on the button. An audio cue instructed subjects to release the button and to reach, grasp, and lift the object to a height marker (5cm) until the next audio cue (4s after button-release time) that instructed them to

return the object and hand to their respective start positions. The start cue of the first trial was aligned with a functional image. An error cue was given after trial completion if the object roll exceeded 5° at any time during the trial. Stimulus timings for each block of trials were controlled by a custom script (Vizard Virtual Reality Software Toolkit, version 4.0, Worldviz), and the inter-trial interval was randomly chosen to be between 2- 6 s, with a rest period between each of the four blocks of trials. Trial order within a given block was counterbalanced across runs and subjects.

Following practice, BOLD contrast was measured as subjects completed 40 trials in each of 6 functional runs (for a total of 240 trials). For each run's fMRI analyses, we parsed these trials in the following way, giving 7 conditions of interest for unconstrained and constrained conditions, respectively:

1. early pre-rotation trials 2-4
2. mid pre-rotation trials 5-7
3. late pre-rotation trials 8-10
4. rotation trial 11
5. early post-rotation trials 12-14
6. mid post-rotation trials 15-17
7. late post-rotation trials 18-20.

4.1.5 Kinematic Data Processing

Kinematic data were filtered using a fourth-order Butterworth filter (cutoff frequency = 5 Hz). We defined object roll as the angle of the object in the frontal plane, with peak object roll extracted shortly after lift onset (250 ms) before somatosensory feedback resulted in corrective responses to counter object roll. Trials with object roll $> 5°$ were

classified as errors. Lift onset was defined as the timepoint when the object was lifted 1mm and remained above this value for at least 20 samples.

4.1.6 MRI Data Preprocessing

MRI data were pre-processed and analyzed in SPM12 (Wellcome Trust Center for Neuroimaging, London, UK). Specifically, functional images across all runs were spatially realigned to a mean functional image using 2nd degree B-spline interpolation, which were then co-registered to each subject's structural T1 image. Between-subject spatial normalization steps were conducted with SPM's normalize function aligning each subject's T1 and its co-registered functional images into standard ICBM/MNI-152 atlas space (interpolation: 4th degree B-spline; voxel size: 3x3x3 mm).

We used a deconvolution-based general linear model (GLM) approach to model BOLD activity, with a finite impulse response (FIR) function selected as a basis function (window length: 6.8 s; order: 400 ms), yielding 17 400 ms time bins. Bins 0 and 1 relate to neural activity present before lift onset; lift onset occurs at the start of bin 3. As described above, for each run, we modeled 7 conditions for unconstrained and constrained trials, respectively, with three pre-rotation conditions containing trials 2-4, 5-7, and 8-10; a rotation condition containing trial 11; and three post-rotation conditions containing trials 12-14, 15-17, and 18-20.

Finally, we used the RobustWLS Toolbox in SPM [88] to account for movement artifact by an unbiased estimation of noise variance of each imaging and down-weighting of images with high variance. Nevertheless, head motion mean rotations and translations (with minimum and maximum values in parentheses) were minimal: x: -.02 mm (-.38,

.34); y: -.29 mm (-.86, .29); z: .76 mm (-.42, 1.72); pitch: -.008° (-.02, .008); roll: -.001° (-.009, .006); yaw: .002 ° (-.005, .01).

Before use in estimating the neural deviation, the BOLD values across different ROI's were aggregated into vectors. Given that the task under consideration was a sensorimotor task, it would be natural to restrict the regions under consideration to sensorimotor regions. We show in Figure 4.1 that this is unnecessary, as the deviations generated by sensorimotor regions (vertical axis) are directly proportional to those generated by global activity (horizontal axis). The red lines demonstrate approximate equivalence: the sample deviations cluster about this line for all conditions. The sensorimotor ROI's selected here were the bilateral anterior intraparietal sulcus (AIPS), the Cerebellum, Insula, motor 4a, motor 4p, parietal operculum, primary somatosensoty cortex, and superior parietal lobule (SPL). Before deviations were computed, the BOLD vectors were mapped to a lower dimensional space (the space used was ten dimensional). A basis for this space was computed using the Treelet Transform [**?**] because of its ability to capture sparse, hierarchical structure in covariance matrices.

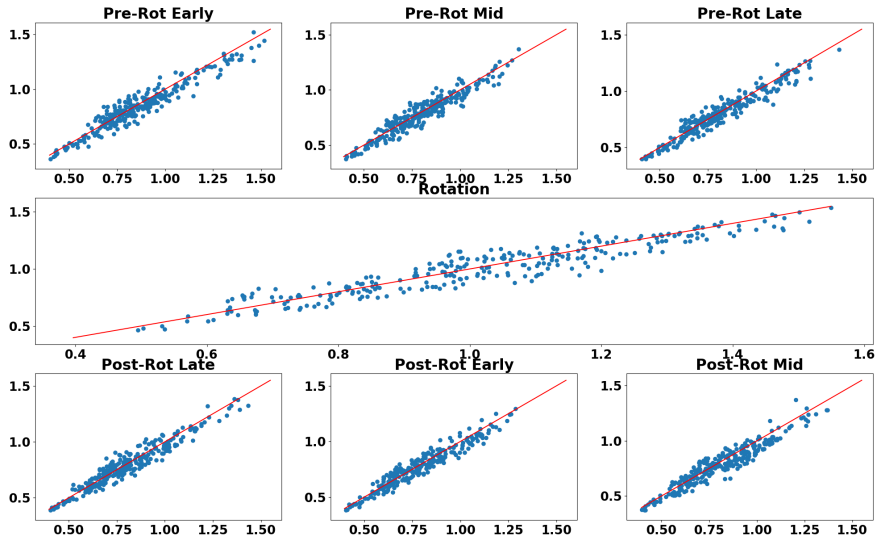


Figure 4.1: All plots are of sensorimotor deviations (vertical axis) against global deviations (horizontal axis). The red line gives perfect equality and the samples cluster about this line.

4.1.7 Robotic Simulation Details

The OpenAI Gym Pick and Place environment was modified to replicate the experimental task described in this chapter. Specifically, the block to be moved was extended along a single axis to allow for shifting of the center of mass of the block along this extended axis. Adapting to lift this unbalanced weight with minimal roll along the extended axis would then test the ability of the robot to perform a similar task to that accomplished by the human subjects. Two prior policies were trained using Deep Deterministic Policy Gradients (DDPG) and Hindsight Experience Replay (HER) to lift the block with minimal roll when its center of mass is centered and uncentered, respectively [89]. The parameters used for training were the defaults given in [89]. A single frame taken of the simulation is shown in Figure 4.2.

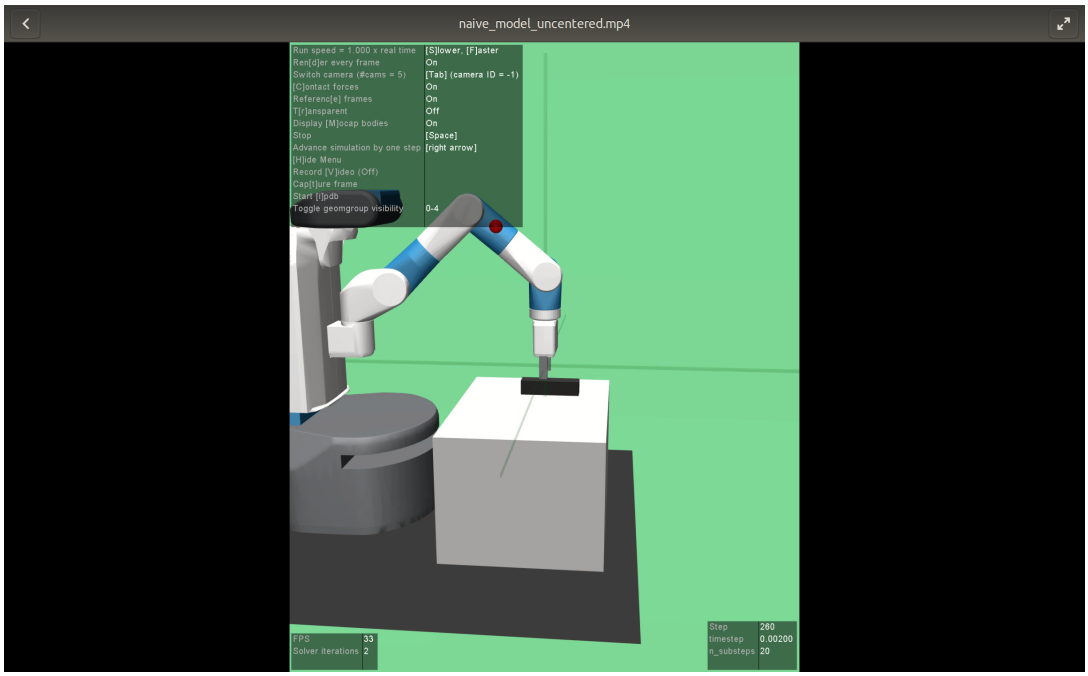


Figure 4.2: A still frame taken from the robotic simulation.

The dynamics model used for Model Predictive Control (MPC) was a deep neural network with 3 layers and 256 neurons per layer. This network was trained using ADAM with learning rate 0.001 and batch size 256 [48]. Mini-batches were sampled from a uniform distribution over elements of the replay buffer, which had a maximum size of 1e6 elements. A zero'th order policy optimization scheme was used within the MPC framework. For this optimization scheme, 500 rollouts were used, each of length 15 timesteps.

4.2 Errors During Motor Learning are Probabilistically Distributed

First, we examine behavioral performance during the adaptation task. Across all participants, trials and conditions, the maximum magnitude roll over the course of a trial,

averaged over all subjects and runs, was observed to be low for pre-rotation conditions, high for rotation conditions, and low again for post-rotation conditions. This point is illustrated in Figure 4.3, where we show the distributions over states for all conditions. These plots not only make clear the presence of errors and the fact that they are quickly corrected, but also that the distributions over states contain meaningful information that would be lost by considering only the mean. For example, the distribution generated by the rotation condition has a different shape from any of those generated from the pre-/post-rotation conditions (pre/post vs rot, A^2=89.21431, p<0.01; throughout this work, pre/post refers to the combination of pre-rotation and post-rotation samples and rot refers to rotation samples). This result holds after bootstrap resampling of samples to correct for sample-size differences between pre-rotation/post-rotation conditions and the rotation condition. Plots of the resampled histograms are shown in the second row of Figure 4.3. The resampled histograms were generated by resampling pre-rotation and post-rotation samples uniformly at random to generate sample sizes equal to that of the rotation condition.

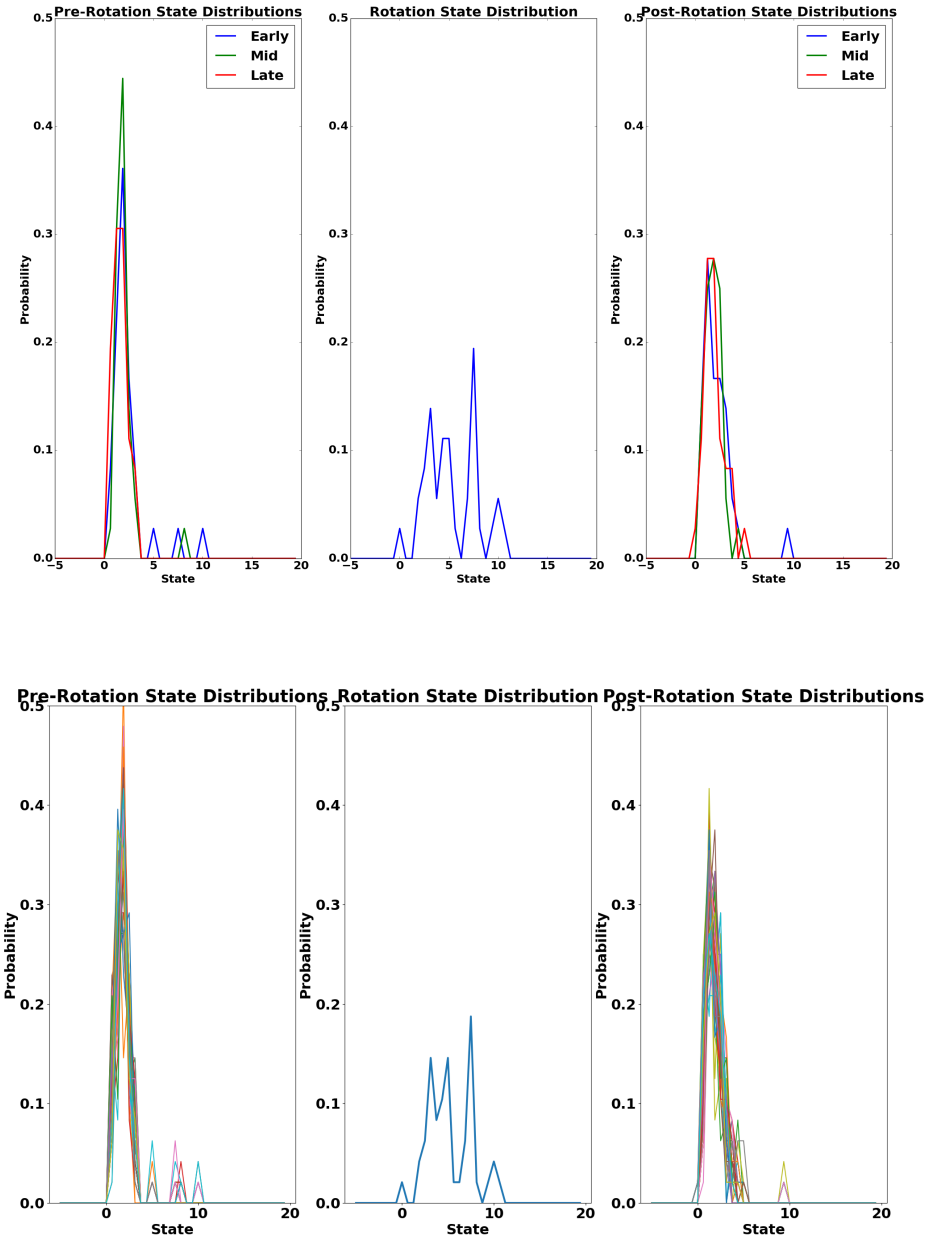


Figure 4.3: **Top**: state distributions. The rotation condition results in a state distribution that has both higher mean and an entirely different shape. **Bottom**: bootstrap resampling of histograms to yield balanced sample sizes across pre-rotation, rotation, and post-rotation conditions results. The difference in the rotation histogram compared with the pre-rotation and post-rotation histograms is preserved even in the case of bootstrap resampling.

Analysis of these results requires a representation of the error that takes into account the observed distributional information. We observed that these distributions over distances follow a Weibull distribution

$$p(d_p; \gamma, \beta) = \frac{\gamma}{\beta}(\frac{d_p}{\beta})^{\gamma-1} e^{-(\frac{d_p}{\beta})^{\gamma}},$$

where d_p is the L_p distance and γ and β are parameters. This provides a convenient, closed-form mathematical representation for errors that we revisit throughout this work. To validate that our distances are indeed Weibull distributed, consider first the necessary and sufficient conditions for distances between feature vectors to be Weibull distributed. Given feature vectors $X = [X_1, \ldots, X_n] \in \mathbb{R}^n$ and $Y = [Y_1, \ldots, Y_n] \in \mathbb{R}^n$, the L_p distance between X and Y is Weibull distributed if $|X_i - Y_i|^p$ are non-identical, correlated, and upper bounded, for all $1 \leq i \leq n$. Rather than construct a mathematical proof that these assumptions hold for human movement, we instead demonstrate that Weibull distributions can be successfully fit to our data.

In Figure 4.4 we show that the empirical distributions over distances resulting from comparing pre/post samples with rotation samples (called rotation or rot) differ significantly from the empirical distributions generated by comparing pre/post samples with other pre/post samples (pre/post-rot vs pre/post-pre/post, A^2=60.66115, p<0.01). Moreover, fitting Weibull distributions to these empirical distributions using Maximum Likelihood Estimation (MLE), we are able to generate accurate fits, suggesting that the Weibull is indeed a good model for these data (pre/post empirical vs pre/post Weibull, β=2.047, γ=1.062, A^2=1.184024, p>0.2; rot empirical vs rot Weibull, β=4.947, γ=1.504, A^2=1.72791, p>0.2). We call the pre/post-rot Weibull the error Weibull (W_e) and we call the pre/post-pre/post Weibull the ideal Weibull (W_i). As subjects adapt and W_e is

transformed back to W_i, a number of characteristics of W_e change: its mean shifts towards 0, its long tail becomes reduced in size, its variance shrinks, and its skew decreases. From this, it seems as if some notion of the deviation between W_e and W_i would have to be used as feedback to a controller in order to incorporate all of this information.

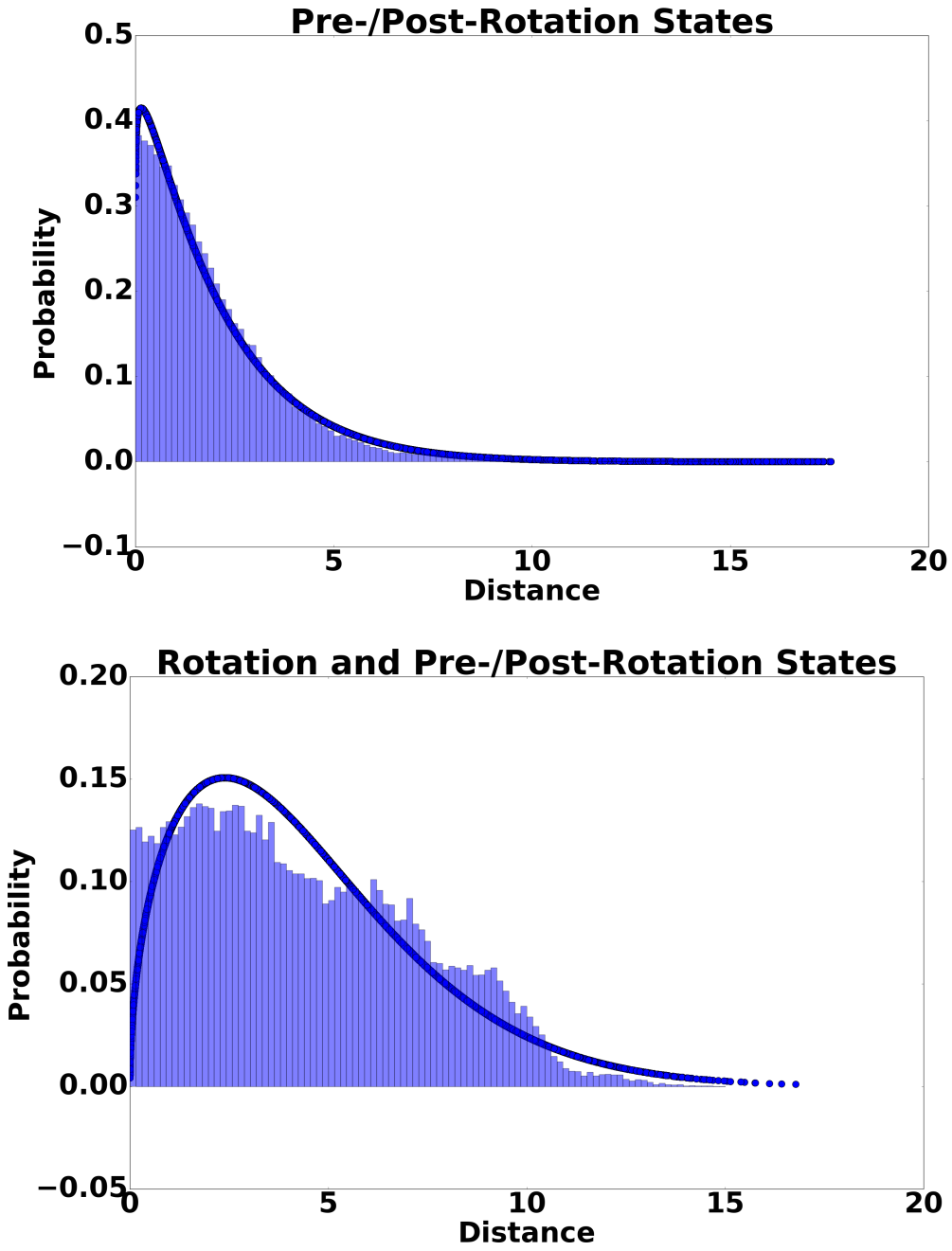


Figure 4.4: The empirical density estimates are presented as histograms and the Weibull fits are superimposed and given by the blue dots. **Left**: distances are between pre-/post-rotation conditions. **Right**: distances are between rotation and pre-/post-rotation conditions.

4.3 A Distributional Model for Prediction Errors

A model of learning that relies on the deviation between W_e and W_i can be derived from a Temporal Difference Model (TDM) [86], which can in turn be derived from a Temporal Difference Learning (TDL) update. TDL is a recursive scheme to maximize expected future rewards and requires the definition of a value function, $V(s_t|\pi)$, where s_t is the state at time t and π is a policy. The value function can be defined as

$$V(s_t|\pi) = \mathrm{E}_{p(s_{t+1}|s_t,a_t),\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots],$$

$$V(s_t|\pi) = \mathrm{E}_{p(s_{t+1}|s_t,a_t),\pi}[r_t + \gamma V(s_{t+1}|\pi)],$$

where $\gamma \in [0, 1)$ is a discount factor, $p(s_{t+1}|s_t, a_t)$ is a model of the system dynamics, $\mathrm{E}[\cdot]$ is the expectation operator, and r_t is the reward at time t. Perhaps the simplest approach to fitting $V(s_t|\pi)$ using TDL, called TD(0), relies on computing an estimator $\hat{V}(s_t|\pi)$ using the update equation

$$\hat{V}(s_t|\pi) = \hat{V}(s_t|\pi) + \alpha[r_t + \gamma\hat{V}(s_{t+1}|\pi) - \hat{V}(s_t|\pi)],$$

where $\alpha \in \mathbb{R}$ is the learning rate. This update involves a comparison between $r_t + \gamma\hat{V}(s_{t+1}|\pi)$ and $\hat{V}(s_t|\pi)$. The intuition for this update is, since the former has slightly more information from the environment than $\hat{V}(s_t|\pi)$, $\hat{V}(s_t|\pi)$ should be updated to be closer to it. TDMs define a reward function using the notion of a goal state, s_g, where $r_t = r(s_t, a_t, s_{t+1}, s_g) = -d_p(s_{t+1}, s_g)$ and d_p is the L_p distance. This reward results in a value function that quantifies the expected future proximity of the system to the goal state. As applied to our experimental system, if we let t and $t+1$ be trial numbers, since the goal state is a roll of zero, $V(s_t|\pi)$ would then indicate the expected magnitude of

the roll over future trials.

TDMs rely on a kind of reward prediction error to update the value function, and can also act as a bridge between state prediction and reward prediction errors. To be clear, state prediction error is the error in predicting the next state given the current state, and reward prediction error is the ability to predict the future reward given the current state. These errors, when applied to our system, quantify the ability to predict future rolls in expectation. This is shown in the Supplement, where we give conditions for the equivalence of state and reward prediction in the TDM framework. In our experimental system, we are not simply interested in defining V using an expectation over $p(s_{t+1}|s_t, a_t)$ and π. We would like to be able to use all of the information contained in the distribution over rewards. To this end, we incorporate TDMs into the Distributional RL framework

$$Z(s_t, a_t, s_g) \stackrel{D}{=} R(s_t, a_t, s_{t+1}, s_g) + \gamma Z(s_{t+1}, a_{t+1}, s_g),$$

where Z and R are the value and reward distributions, respectively, and $\stackrel{D}{=}$ indicates equality in distribution [90]. Similar to TD(0), Distributional RL updates an estimator of the value distribution, $\hat{Z}(s_t, a_t, s_g)$, by comparing $R(s_t, a_t, s_{t+1}, s_g) + \gamma\hat{Z}(s_{t+1}, a_{t+1}, s_g)$ with $\hat{Z}(s_t, a_t, s_g)$. Because these are probability distributions, $\hat{Z}(s_t, a_t, s_g)$ is updated to minimize

$$D_{KL}(R(s_t, a_t, s_{t+1}, s_g) + \gamma\hat{Z}(s_{t+1}, a_{t+1}, s_g)||\hat{Z}(s_t, a_t, s_g)),$$

where $D_{KL}(\cdot||\cdot)$ is the KL-divergence. This update is analogous to the temporal difference learning update, generalized to the setting where rewards are probabilistically distributed. This distributional objective, with $R(s_t, a_t, s_{t+1}, s_g)$ defined as the distribution over $-d_p(s_{t+1}, s_g)$, is relevant in the context of the results presented thus

far. Specifically, in the case of short-time horizon problems, those where $\gamma = 0$, then $Z(s_t, a_t, s_g) \stackrel{D}{=} R(s_t, a_t, s_{t+1}, s_g)$ follows a Weibull distribution.

Keeping with the notation of the previous section, we can think of $\hat{Z}(s_t, a_t, s_g)$ as being equivalent to W_i during the pre-rotation conditions. When the center of mass changes, $Z(s_t, a_t, s_g)$ is actually W_e, though $\hat{Z}(s_t, a_t, s_g)$ is still W_h. The deviation between $\hat{Z}(s_t, a_t, s_g)$ and $Z(s_t, a_t, s_g)$, that is, W_e and W_i, is optimized during adaptation. For the experimental system studied in this work, there are a number of potential explanations for this deviation, from errors in the model of system dynamics to errors in the behavioral policy. The identification of the precise source of the deviation between $\hat{Z}(s_t, a_t, s_g)$ and $Z(s_t, a_t, s_g)$ is beyond the scope of this work. Our goal is to present a framework for modeling learning with stochastic rewards in a manner amenable to both biological modeling and robotic control. With this in mind, we note that $\hat{Z}(s_t, a_t, s_g)$ may be parameterized by θ, which includes parameters for every component of the controller used to solve the unbalanced lifting task. We can now propose a model for motor learning, specifically, a model for learning to dynamically update a controller to lift an object in response to its changing physical properties. Our model is that the brain attempts to solve the following optimization problem

$$\underset{\theta}{\text{minimize}} \; D_{KL}(W_i || W_e), \tag{4.1}$$

This is a special case of the full DTDM optimization, but throughout the rest of the chapter, when we refer to the DTDM problem, we are referring to Equation 4.1.

4.4 Global Neural Activity Optimizes the Distributional Temporal Difference Objective

We have already shown that behavior is updated to optimize the deviation between W_e and W_i, that is, behavior is updated according to Equation 4.1. To see the effect of the object rotation condition on global neural activity, we first processed brain activity in consecutive time intervals using finite impulse response (FIR) modeling. We then selected FIR time bins that are likely encoding information about the lift of the apparatus. Details of the method used to select the "lift" FIR bins are given in the Supplement. Briefly, we first identify "pre-lift" FIR bins as those before lift onset: this occurs at FIR bin 3. We then interpret the hemodynamic response as a stochastic process and note that there are two distinct stimuli within each trial: the pre-lift and lift stimuli. Given that these stimuli are separated in time, their respective hemodynamic responses will peak at different times. This allows for the segmentation of the FIR bins as most likely generated from either the pre-lift or the lift process. Those most likely generated from the lift process (bins 15-16) are called "lift" bins and are used to estimate the deviation of global neural activity resulting from lift. These bins are identified using a hard threshold based upon a model of the hemodynamic response (i.e. the Canonical Hemodynamic Response Function, CHRF) [91]. We interpret the CHRF as a mixture of Gamma Distributions. Using two CHRF's (one corresponding to pre-lift and one corresponding to post-lift), we are able to segment the FIR bins as most likely exhibiting BOLD activity from pre-lift or post-lift. Further details on this method are given in the Supplement.

These results are shown in Figure 4.5. For each condition (pre-rotation early/mid/late, rotation, post-rotation early/mid/late), Weibull distributions were generated by comparing the betas generated during that condition with the betas generated during all others.

Example Weibull distributions generated during FIR bin 0 and FIR bin 15 are shown in the left and right columns of the top two rows of Figure 4.5. The distribution generated using the rotation condition exhibits a significant deviation from the others at FIR bin 15 but not FIR bin 0. Because pre/post-pre/post and pre/post-rot Weibull distributions are statistically different for lift bins but not pre-lift bins (pre/post-rot vs pre/post-pre/post for pre-lift bins, t(df)=-1.572965, $p>0.2$; pre/post-rot vs pre/post-pre/post for lift bins, t(df)=-8.73572, $p<0.01$), this suggests that global neural activity is perturbed by the rotation condition, and then moves back to become indistinguishable from the pre-rotation state. Thus we call the pre/post-pre/post Weibulls "ideal beta Weibull distributions" or W_i^b and the pre/post-rot Weibull the "error beta Weibull" or W_e^b.

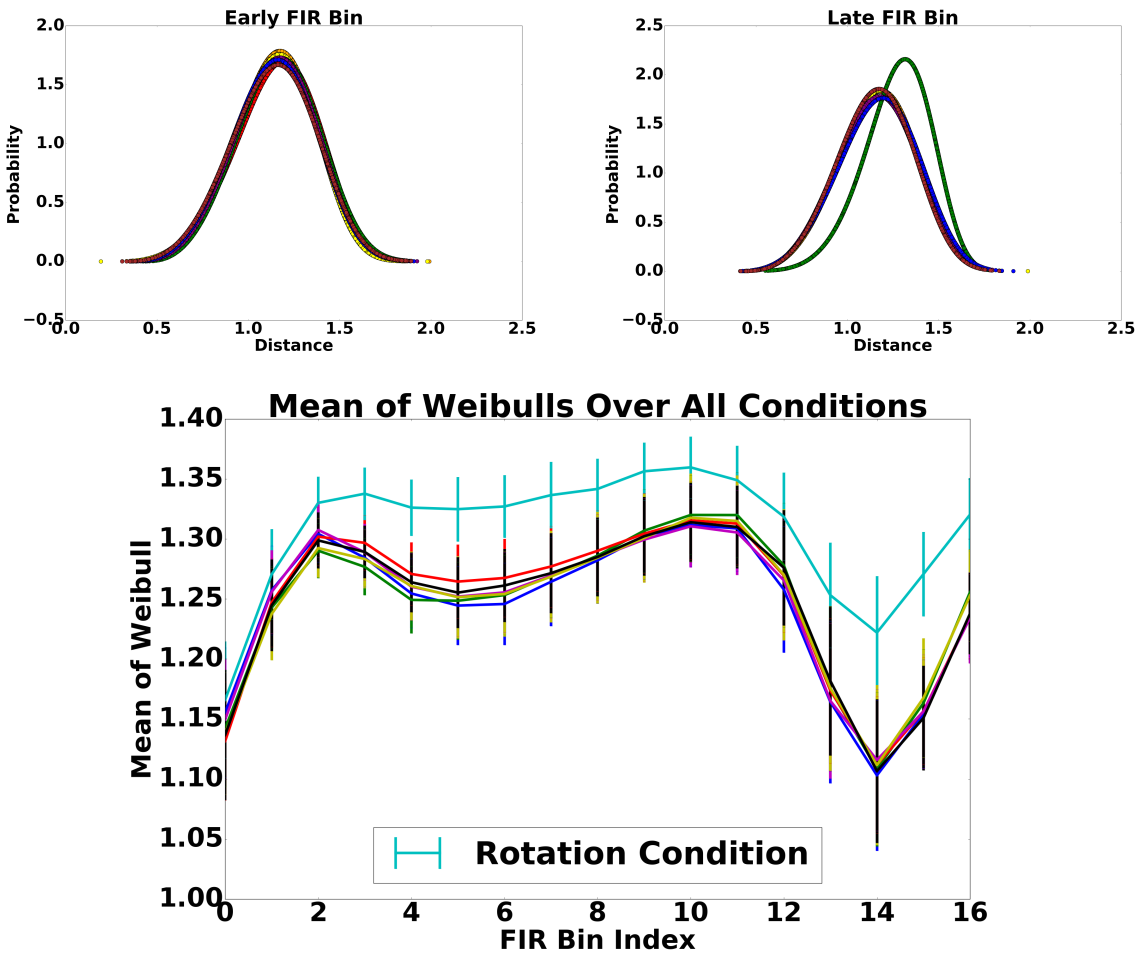


Figure 4.5: The **top** row contains the fits generated using FIR bins 0 and 15. W_h^b becomes distinguishable from W_e^b around FIR bin 15. This point is illustrated in the **bottom** row. This row contains the estimates of the means of the Weibull distribution for each condition against the FIR bin index. The significant deviation of W_h^b from W_e^b for late FIR bins is captured by these plots.

Our results suggest that the brain may be sensitive to $D_{KL}(W_i||W_e)$. In Figure 4.6 (bottom row) we show that the difference in the means of W_i^b and W_e^b (using lift FIR bins) is directly proportional to the deviation between W_i and W_e (i.e. $D_{KL}(W_h||W_e)$; $R^2 = 0.55$). We show in Figure 4.6 (left, middle row) that global neural activity is also directly proportional to the TDM error, that is, errors in expected future reward ($R^2 = 0.44$). To understand this result, we present histograms estimating W_i and W_e

from two representative subjects. The transport of W_e to W_i involves more than just a shift in the mean for both subjects, but for both (and for all other subjects as well), the mean is indeed shifted during adaptation.

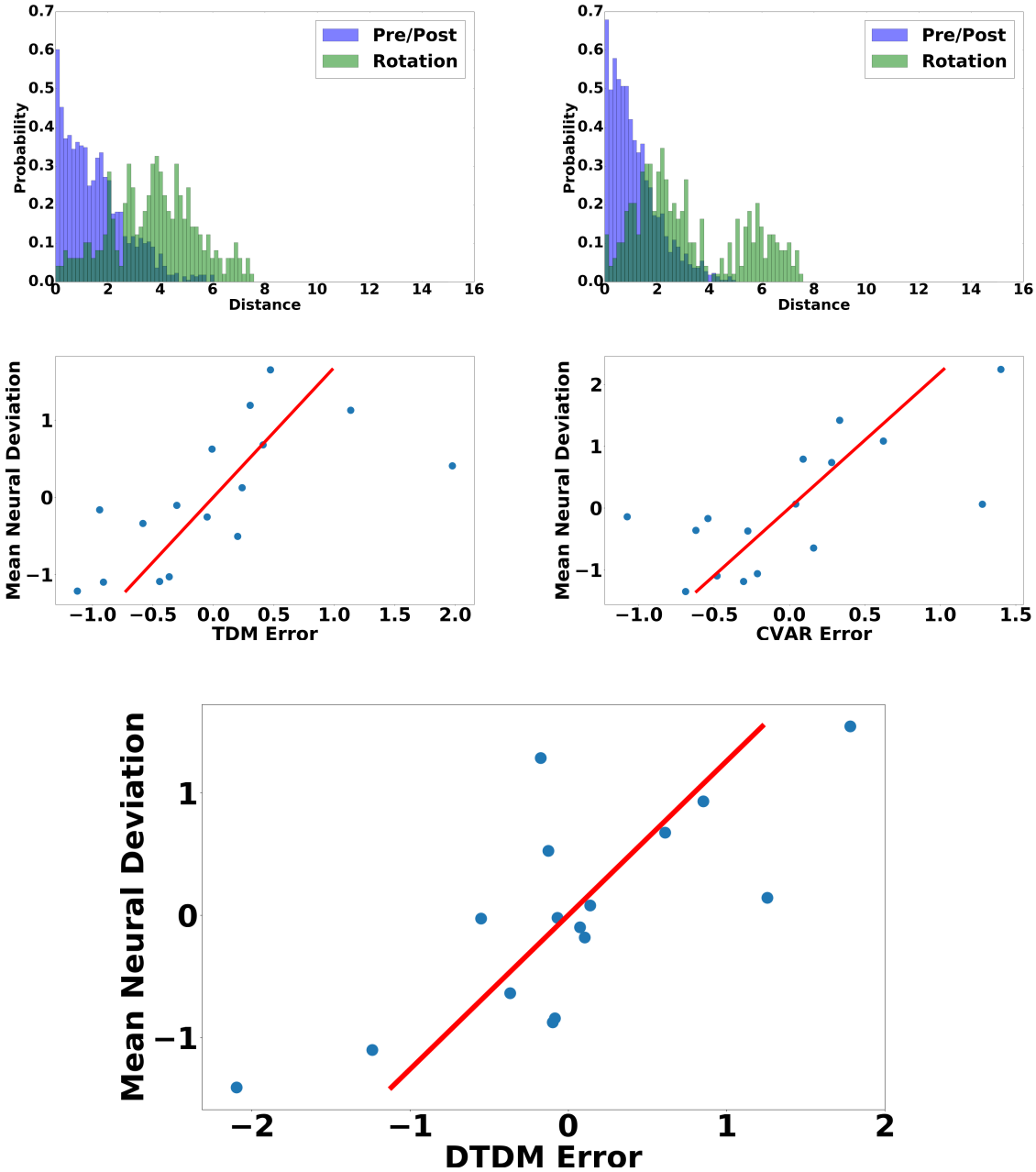


Figure 4.6: The **top** row contains histograms of pre-/post-rotation and pre-/post-rotation with rotation state distances generated from two subjects. Modeling the movement of the mean of the rotation distribution is not sufficient to completely characterize the learning objective. The **middle** row relates TDM and CVAR errors with the mean beta deviation, where each point is a subject. The **bottom** plot illustrates the relationship between the DTDM error and the mean beta deviation. Lines of best fit are shown in red and are generated using the RANSAC algorithm because of its robustness to outliers. Arguably, the mean neural deviation is encoding both TDM and CVAR errors (as well as other relationships between W_h and W_e).

It is important to note that TDM does not contain a complete description of the errors. To see this quantitatively, we use a Conditional Value at Risk (CVAR) model [92]. CVAR models offer a means of taking advantage of the information contained in the value distribution, beyond its mean. These models involve optimizing the expected value in the tails of the value distribution. For example, minimizing lower tail values results in controllers that are risk averse. Risk aversion in our experimental system would involve minimizing the use of actions leading to outcomes in the tail of W_i. For example, suppose that subjects initially used lifting strategies that sometimes led to states near zero (extremely successful outcomes), but also often led to the apparatus being dropped, resulting in high roll. A risk averse learning process would avoid this strategy, leading to fewer observations in the tail of W_i. Interestingly, because this may also reduce the observation of as many low roll states, the mean of W_e may be unaffected by risk averse learning. We show in Figure 4.6 that the CVAR error (i.e. the expected lower-tail value) is also proportional to mean neural deviation ($R^2 = 0.46$). Because CVAR error is a characteristic of the value distribution and is independent of the means of W_i and W_e, this suggests that the global neural deviation is, in fact, also encoding more than just the expected future reward.

The error $D_{KL}(W_i||W_e)$ can be interpreted in a number of ways since different aspects of neural activity could contribute to this shift. Sensory activity as well as error signaling could contribute to such a shift. In addition, compensatory behaviors were also observed during the course of a lift. When a subject perceived a tilt, they would often attempt to change the forces and torques used during the course of the lift, often resulting in reduced roll. We hypothesize that there exists a coordinated, global response to errors that incorporates all of this information and that it is proportional to $D_{KL}(W_i||W_e)$. To show that the global shift in neural activity can be directly used as a feedback error

signal, we use $D_{KL}(W_i||W_e)$ to fit a robotic controller.

4.5 Robots Can Optimize the Distributional Temporal Difference Objective

Conveniently, the optimization problem in Equation 4.1 leads to a form that can be optimized by an artificial agent. To see this, we consider an optimization problem similar to those used to update models of system dynamics for use in Model-Based RL. A popular objective for fitting a model of system dynamics is

$$\underset{\theta}{\text{minimize}}||s^*_{t+1} - f_\theta(s_t, a_t)||_2^2, \tag{4.2}$$

where f_θ is a model parameterized by θ, s^*_{t+1} is the true state at time $t+1$, and $f_\theta(s_t, a_t)$ is the predicted state at $t+1$. On its face, it may not be obvious how Equation 4.2 is related to Equation 4.1. The latter involves fitting W_e, which is a distribution over distances between the current and target states, while the former involves comparing predicted and actual states at time $t+1$. To see the connection, consider the fact that if f_θ is a probabilistic model, even if its performance is optimized via Equation 4.2 (with some steps taken to preserve non-zero variance), the distances $||s^*_{t+1} - f_\theta(s_t, a_t)||_2$ will be Weibull distributed. We can think of this Weibull as W_i. In the case where the environment changes and the state at time $t+1$ is no longer s^*_{t+1} but instead $s^{'}_{t+1}$, the performance of f_θ is no longer measured by $||s^*_{t+1} - f_\theta(s_t, a_t)||_2$. Instead, $||s^{'}_{t+1} - f_\theta(s_t, a_t)||_2$ is used. The distribution over these new distances is no longer W_i, and we call this new Weibull W_e. Updating the dynamics model using Equation 4.1 would then amount to bringing the predictions of $f_\theta(s_t, a_t)$ as close to $s^{'}_{t+1}$ as they had been to s^*_{t+1} before the environment

changed. We incorporate Equation 4.1 into a model-based RL approach. We use this model-based framework to allow a simulated robotic arm to learn to lift a block when the location of its center of mass is periodically shifted.

The controller we use assumes the existence of two stochastic policies: one that is capable of lifting an object with a centered center of mass and another that is capable of lifting an object with an unbalanced center of mass. We make this assumption because in learning to adapt to a shifting center of mass, the human subjects in our experiment already know how to lift the object in both orientations. The task is assessing their ability to adapt, thus this is the focus of our robotic experiment as well. At time t of the simulation, R possible actions are sampled from the policies. Rollouts from these actions are simulated forward in time to $t+T$ using a dynamics model and the policies. This results in R state-action trajectories of length T. These trajectories are compared using the cumulative reward over all T timesteps, $\sum_{h=t}^{t+T} c(s_h^i, a_h^i)$, where $i \in \{1, \ldots, R\}$ and $c(s_h^i, a_h^i)$ is the absolute value of the roll of the object at time h. The action at time t yielding the lowest cost trajectory is the one selected and this process is repeated for each timestep.

The results of this experiment are shown in Figure 4.7. The top plot shows the error generated by the dynamics model with respect to the trial number. Shortly after trial 400, the center of mass is switched, causing a spike in the error. Within about 50 trials, the model has adapted and its performance has improved to be better than it was before the switch. The bottom plot shows the performance of the controller as measured by the absolute value of the roll over the course of the trial. The results show that the robot is able to adapt quickly to the changing center of mass, albeit not as quickly as a human. The robot is able to adapt in a little over 100 trials, while the human is able to adapt

within 1-2 trials.

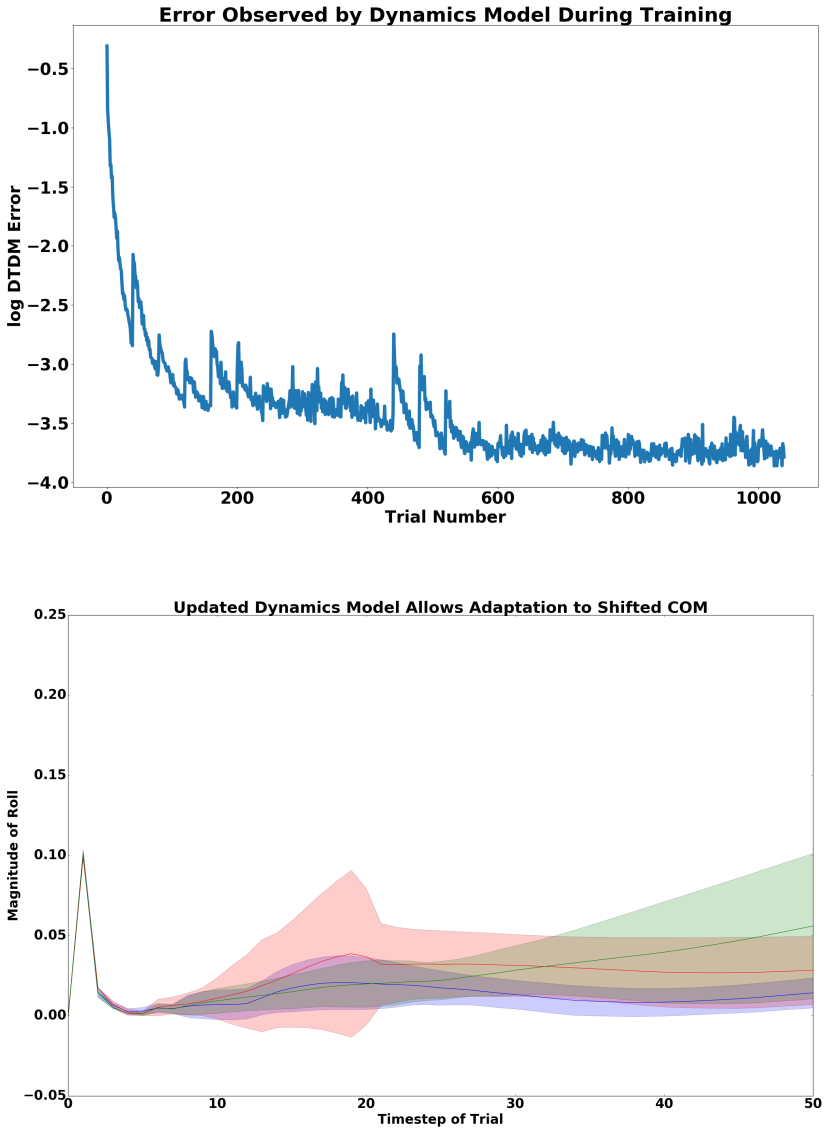


Figure 4.7: **Top**: Error of the dynamics model against the number of iterations of training algorithm. Training was performed using mini-batch sampling, which explains the saw-tooth improvement in the error. The center of mass is switched half-way through training, resulting in a jump in the error. This jump is corrected during subsequent updating. **Bottom**: The **red** curves are generated by the controller fit using a centered weight with dynamics trained on a centered weight, the **green** curves are generated using an uncentered weight with a dynamics model trained on a centered weight and not updated, and the **blue** curves are generated using an uncentered weight with an updated model. The updated model is able to outperform the controller without a model update.

There are many possible sources of inefficiency for the robotic controller that could explain this performance gap. First, the dynamics model is updated using random batch sampling from past experience. Arguably, humans don't randomly sample from all past experiences with the object when faced with sub-optimal performance. They are able to draw from past experience based upon hypotheses as to the cause of the errors. Next, the dynamics model is relatively simple and contains no prior knowledge before training about how such objects behave. The human mind contains an enormous amount of past experience to draw on to generate hypotheses explaining errors. The representation of the object in the human mind is also much higher dimensional than the representation used by the robot, containing tactile, visual, and auditory information. Yet, while the performance of the robotic controller is not at the level of the human, these experiments demonstrate that the DTDM objective can actually be used to solve a control problem that is similar to the one solved by humans.

4.6 Discussion

We have proposed a distributional learning objective for use during motor control and used this representation to construct a model of motor learning. To so do, we extended Temporal Difference Models to Distributional Temporal Difference Models. We have shown that behavior appears to optimize this distributional objective and that deviations in global neural activity are proportional to the magnitude of the distortion of the value distribution. DTDM is not simply useful as a model of motor learning. We have shown that it can be incorporated into a robotic controller and used for engineering applications. The strong connection implied between neural and robotic systems suggests that improved understanding of the brain can be directly used to improve robotic engineering. Our work also suggests that work exploring the converse claim may be successful

as well. This claim is often made indirectly, for example, by citing the neuroscientific origins of machine learning, though there is currently no formal framework for extracting neuroscientific principles for the purpose of engineering AI [79]. We hope that this work will be a step in this direction.

Our results also contribute to the accumulating body of evidence in support of the Minimum Free Energy (MFE) theory of neural learning [67, 66, 70, 69, 68, 64, 63, 62, 65, 60, 85]. This theory posits that learning proceeds through the optimization of a free energy of the form $\mathrm{E}_\rho[V] + \mathrm{H}[\rho]$, where V is a potential function, $\mathrm{E}[\cdot]$ is the expectation operator, and ρ is a probability measure. Interestingly, optimization of this free energy is equivalent to optimization of $\mathrm{D}_{KL}(\rho||e^{-V})$ [85]. In this work, we have not explored the extent to which W_i can be approximated by a measure of the form e^{-V}: this information would allow for Equation 4.1 to be directly related to a free energy functional. This may be an interesting direction for future work. In developing a model as well as a controller using the idea of KL-control, we have contributed another set of results that strengthen the claims of the MFE theory.

We have proposed a distributional framework for motor learning, but haven't explored in depth how different aspects of the value distribution could be used during motor adaptation. In Figure 4.4, we show that W_e has a much longer tail than W_i in addition to having a higher mean, and that these tail values are also optimized during adaptation. Using the difference in the means of W_e and W_i obscures this information, despite the fact that it is useful in a number of different settings. Certainly, there are situations in which risk-averse behavior is best and situations where it results in overly cautious behavior. By maintaining a representation of the value distribution, the brain is able to generate policies by optimizing different aspects of the distribution. These policies can

be selected from, to produce behavior that is appropriately cautious for a given situation.

The issue of selecting from amongst a population of possible actions is interesting in the context of DTDM for other reasons as well. Often, the representation of errors used in control problems and in modeling neural controllers is subordinate to the type of controller used, for example, either model-based or model-free. This work suggests that from both a neurological and an engineering standpoint, this manner of thinking may be reversed. Specifically, it may be better to develop a representation of errors that can be used for either model-free or model-based control, and then develop a controller that can best optimize this error in the system of interest. In the context of neurological systems, this suggests the existence of a generic error encoding that are independent of the class of controller. The utility of such a generic error representation would facilitate, for example, action selection in the setting where a number of candidate actions must be selected from and the candidate actions are generated from both model-free and model-based systems [93, 94, 95, 96, 97]. In this setting, a generic representation of error would allow for a universal way of comparing the performance of controllers and selecting actions.

Chapter 5

Human Learning and Multi-Objective Reinforcement Learning

Development of reinforcement learning methods for complex tasks requires selection of a suitable reward function, inference on the structure of this reward function, or some combination of these two approaches. Inverse methods which allow for reward functions to be inferred from optimal demonstration data are often an effective alternative to manual engineering because of the difficulty of defining an unambiguous reward function over large state-action spaces. Existing Inverse Reinforcement Learning (IRL) methods assume that rewards are scalars, an assumption which is arguably inappropriate for complex tasks which are often characterized by many distinct, correlated reward functions. We call this the Multi-Objective IRL (MOIRL) problem in this work (as opposed to existing Scalar IRL (SIRL)), and describe a novel approach for solving it.

The problem of solving RL problems for complex tasks has been studied in many dif-

ferent ways. One solution, in the case where rewards need to be manually engineered, is reward shaping [98, 99, 29]. Reward shaping involves augmenting a complex reward function with simpler ones. These simple rewards are more easily optimized and, intuitively, may correspond to components of complex tasks (e.g. grasping a block before attempting to lift it). Hierarchical RL takes a similar approach, where complex tasks are decomposed into simpler tasks by helping an agent to navigate between goal states on the way to performing the complete task [100, 101]. An agent is motivated to move between goal states using auxiliary reward functions which are easier to optimize than the reward for the complete task. These approaches are useful but their generalization to the multi-objective setting or application to the problem of MOIRL is not clear. Our approach solves an inverse problem on the full multivariate reward space, using optimal behavior to avoid the work of manual engineering or fitting of a hierarchical policy.

The use of inverse methods for multiple objectives presents a number of complications over the use of SIRL. In the case where rewards are in the interval $[-1, 1]$, we would like to find a policy which achieves an average reward as close to 1 as possible. In the case of n correlated rewards, finding a policy that is close to the vector of n 1's on average is not possible. To address this problem, there have been a number of attempts to use multi-objective methods in Bayesian optimization [102, 103, 104, 105] as well as reinforcement learning [26, 106, 27, 28, 107]. Existing approaches to multi-objective RL rely largely on linear reward scalarization, which involves projecting the multiple rewards to a single reward function. This may be done with a linear projector that is fixed *a priori* or one that is learned dynamically from interactions with the environment. We argue that linear scalarization is not necessarily appropriate in the setting of IRL. For example, one of the limitations of linear scalarization is that it is impossible to map each point of the multi-objective space to a scalar reward in a one-to-one fashion. Thus, in the setting of

IRL, even if the SIRL problem can be solved, it will be impossible to recover an estimate of the multi-objective reward in general. Because of these complications, the utility of linear scalarization for MOIRL is unclear.

To approach the MOIRL problem, we define optimality (of the demonstration data) in terms of a Pareto set of policy parameters. That is, if the policy that generated our behavior belongs to a Pareto set, then it satisfies a particular necessary condition for optimality. Specifically, members of the Pareto set induce a linear combination of the n objective gradients to sum to 0 (rather than requiring the single objective gradient to sum to 0, as in SIRL) [108, 109]. In addition, this implies the existence of a gradient-based descent direction for the satisfaction of this optimality condition [108, 109]. We show that at steady-state of the policy optimization process, these descent directions each index a different trajectory through the space of multiple objectives. That is, Pareto optimality induces a particular target region of the multi-objective space as well as the dynamics towards this region over the course of a trial. This is distinct from SIRL, where at optimality there is only one reward trajectory and it is completely specified by the reward function. The goal of SIRL is primarily to fit the scalar reward, which allows for policy optimization of this reward. In contrast, MOIRL requires both knowledge of the multiple rewards as well as the descent direction taken during the policy optimization procedure used to generate the optimal demonstrations.

The preceding requirements present a problem in the case where we don't have access to the optimal parameters and to the gradients of the multiple objectives with respect to these parameters. When the optimal demonstrations are human behaviors, this information will generally be unavailable. This makes the use of existing methods for inverting multi-objective optimization impossible [108]. We address this issue by instead assum-

ing that our optimal demonstrations consist of a dataset of optimal state-action-reward trajectories. With this information, we are able to compute a policy that belongs to the extended Pareto set without explicit knowledge of the objective gradients. To accomplish this, we fit a model of the dynamics through the multivariate reward space during each trial. We fit this model of multiple rewards using a deep neural network regressor and show that it has good performance across a sample of human subjects performing a novel visually guided motor planning task with multiple rewards. This suggests that there is a consistent notion of optimality across the population of subjects, and that policy optimization to reproduce the multivariate reward dynamics should be successful. We then apply this approach to an artificial agent, where we can show that with this reward model, we are able to perform model-based policy optimization which imitates the reward trajectories generated by the model. Our results show that the inferred policy of the agent is able to perform comparably to an average human on the planning task, demonstrating the success of our method in inverting human behavior in the multi-objective case.

5.1 Multi-Objective Inverse Reinforcement Learning

5.1.1 Problem Statement

In IRL, the problem of SIRL can be formalized as follows:

$$\max_{\phi} \mathrm{E}_{\tau \sim \mathcal{D}_o}[\mathcal{L}(\tau|\phi)], \tag{5.1}$$

where $\mathcal{D}_o$ is a dataset of optimal state-action trajectories, $\{\tau_o^1, \ldots, \tau_o^n\}$, and $\mathcal{L}(\tau|\phi) = \sum_{t=t'}^{t+T} r_\phi(s_t, a_t)$, where the reward function r_ϕ is parameterized by ϕ. Solving the optimization problem in Equation 5.1 yields ϕ^*. Then policy optimization can be performed directly on $\mathrm{E}[\mathcal{L}(\tau|\phi^*)]$, yielding θ^*. A consequence of this is that, for a fixed $\mathcal{D}_o$ and

optimization algorithm, the reward trajectories generated by θ^* for a single trial are completely determined by ϕ^*.

The situation is more complex when there are multiple objectives. Specifically, suppose we have the following policy parameter update during learning

$$\theta_{k+1} = \theta_k + w_\psi(\mathbf{G}(\tau, k)), \tag{5.2}$$

where $\mathbf{G}(\tau, k)$ is given by

$$\mathbf{G}(\tau, k) = (\nabla_\theta \mathrm{E}[\mathcal{L}_1(\tau|\theta_k)], \ldots, \nabla_\theta \mathrm{E}[\mathcal{L}_n(\tau|\theta_k)]),$$

and ψ are the parameters which ultimately lead to Equation 1.9 being satisfied. Now, suppose that $\theta^*(k)$ is a solution to the dynamics in Equation 5.2. At time k, $\theta^*(k)$ generates the trajectory τ, resulting in the following point in the multivariate objective space:

$$\mathbf{L}(\tau, k) = (\mathrm{E}[\mathcal{L}_1(\tau|\theta^*(k))], \ldots, \mathrm{E}[\mathcal{L}_n(\tau|\theta^*(k))]).$$

Simulating an entire trial using $\theta^*(k)$ yields the sequence of trajectories $[\tau_k^1, \ldots, \tau_k^T]$, resulting in the following path through the multivariate objective space:

$$\mathbf{L}(k) = [\mathbf{L}(\tau_k^1, k), \ldots, \mathbf{L}(\tau_k^T, k)].$$

At time $k+1$, $\mathbf{L}(\tau_{k+1}^t, k+1)$ is given by

$$\mathbf{L}(\tau_{k+1}^t, k+1) = (\mathrm{E}[\mathcal{L}_1(\tau_{k+1}^t|\theta^*(k)+w_\psi(\mathbf{G}(\tau_k^t,k)))],\ldots,\mathrm{E}[\mathcal{L}_n(\tau_{k+1}^t|\theta^*(k)+w_\psi(\mathbf{G}(\tau_k^t,k)))]). \tag{5.3}$$

From Equation 5.3, we can see that at a given time, the different paths through the multivariate objective space are indexed by ψ. In the case where we have demonstrations from $\theta^*(k)$ in the form of state-action trajectories, there are many possible corresponding $\mathbf{L}(k)$, each indexed by a different ψ. Knowledge of the parameters of the reward function ϕ is not sufficient for policy optimization: we must also account for the effect of different ψ's. This is distinct from the problem of SIRL, where knowledge of ϕ is sufficient to perform policy optimization. Here, for a given ϕ there are multiple paths through multi-objective space which may be Pareto optimal. The problem of MOIRL thus requires that we account for the influence of ϕ and ψ when performing policy optimization over multiple, correlated objectives. In order to formalize the MOIRL problem, we require that the dataset $\mathcal{D}_o$ be generated after $\theta^*(k)$ has converged to its steady state: we refer to this time as k_s. In this case, the MOIRL problem can be written as

$$\max_{\phi,\psi} \mathbf{L}(\tau, k_s|\phi) \tag{5.4}$$

$$\text{s.t. } w_\psi(\nabla_{\theta^*}\mathbf{L}(\tau, k_s|\phi)) = \mathbf{0},$$

where $\tau \sim \mathcal{D}_o$. Solving this problem requires knowledge of θ^* or $\nabla_{\theta^*}\mathbf{L}(\tau, k_s|\phi)$, in addition to $\mathcal{D}_o$. Often, we only have access to the samples in $\mathcal{D}_o$ generated using θ^*: this is the case when we only have access to demonstrations from human subjects. In the next section, we propose a way to solve the MOIRL problem given only demonstrations of near-optimal behavior.

5.1.2 Proposed Solution

To address the optimization problem in Equation 5.4, we note that if we have samples along the path $\mathbf{L}(k_s)$, then by assumption, $w_\psi(\nabla_{\theta^*}\mathbf{L}(\tau, k_s|\phi)) = \mathbf{0}$ at these points, satisfying the necessary optimality condition. That is, imitation of samples from $\mathbf{L}(k_s)$ is necessary for a policy θ to belong to the Pareto set, but not sufficient. Policies that satisfy the necessary condition are said to belong to the extended Pareto set (this is also known as the Pareto Critical Set or the Substationary Set) [108, 109]. We introduce a method that imitates the paths $\mathbf{L}(k_s)$ by first building a model of these paths, and then during policy optimization attempts to produce reward trajectories close to those generated by the model.

First, we consider the problem of finding a map that follows $\mathbf{L}(\tau, k_s)$ in expectation, given a state-action trajectory τ as input. Thus the problem becomes

$$\min_{\xi} \mathrm{E}_{\tau\sim\mathcal{D}_o}[||\mathcal{L}(\tau|\xi) - \mathcal{L}(\tau)||_2], \tag{5.5}$$

where $\mathcal{L}(\tau)$ is given by

$$\mathcal{L}(\tau) = (\mathcal{L}_1(\tau), \ldots, \mathcal{L}_n(\tau)),$$

and where ξ are the parameters of the reward model. In order to fit this model, we require that the dataset, $\mathcal{D}_o$ be augmented to consist of state-action-reward trajectories. Here each trajectory is given by $\tau = \{(s_{t'}, a_{t'}, r^1_{t'}, \ldots, r^n_{t'}), \ldots, (s_{t'+T}, a_{t'+T}, r^1_{t'+T}, \ldots, r^n_{t'+T})\}$. With these state-action-reward trajectories, we can compute the ground-truth reward output, $\mathcal{L}(\tau)$, and use the state-action pairs as input to our reward model. The objective

that we optimize is then

$$\min_{\xi} \frac{1}{m} \sum_{i=1}^{m} ||\mathcal{L}(\tau|\xi) - \mathcal{L}(\tau)||_2, \tag{5.6}$$

where m is the number of state-action-reward trajectories in $\mathcal{D}_o$. In this work, we refer to this strategy as reward imitation. In the next section we describe a potential implementation of a reward model, show its success on a novel motor planning task, and describe how it can be used in a model-based RL framework to generate human-level performance on the motor planning task.

5.2 Experimental Validation

5.2.1 Motor Planning Task

The task developed was an extension of the classic grid-sail task [110, 111]. Subjects were given the task of navigating a boat from an initial dock location to a target dock location on a 2D computer screen. To perform this planning task, subjects were given a four element action space: there were three cardinal directions of movement along with the additional null action (no movement; see Figure 5.1 for an illustration). Movement along each of the three cardinal directions could be achieved by pressing one of three keys on a keyboard, while the null action could be achieved by doing nothing.

The task was initialized at the beginning of each trial with a random initial and target dock location. For every trial, each boat was given a fixed amount of gas to be consumed (360 units of gas). Each time an action was performed (excluding the null action) a unit of gas was consumed. At the end of successful trials, subjects were informed of their gas consumption and for unsuccessful trials, subjects were notified why that trial was a

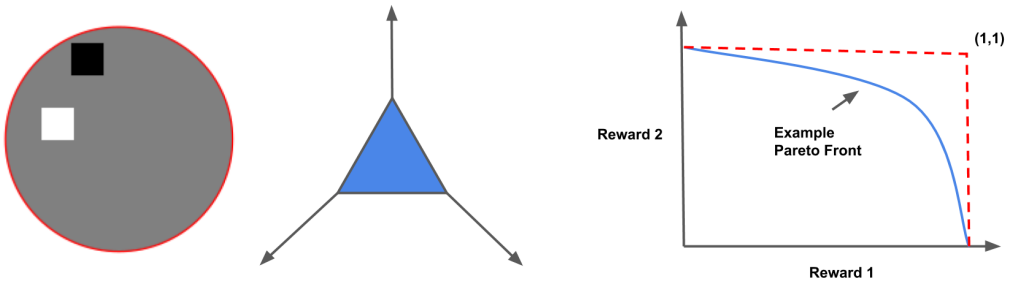


Figure 5.1: **Left**: Summary of boatdock task. On the left is an example of the environment shown to subjects. The initial dock location is shown in white and the target dock is shown in black. On the right is the boat with action space superimposed (excluding the null action). **Right**: Schematic of the Pareto front in the case of two rewards. This curve illustrates the tradeoff between two correlated rewards in a case where both must be simultaneously optimized.

failure (e.g. they used all their gas before reaching the target). During the experiment, subjects were given the choice between either of two action spaces: in the compatible case the cardinal movement directions of the boat were spatially compatible with the arrangement of the fingers on the keyboard. In the incompatible case the cardinal directions were not spatially compatible with the finger to keyboard mapping. For a given trial, only one of the two action spaces offered an optimal trajectory. As a first step in developing an RL method capable of learning optimal behavior in the multi-objective setting, we expose only a single action space to the artificial agent.

Given that the initial and target dock locations were not necessarily collinear with one of the cardinal movement directions, one of the challenges of the task was to move to the target efficiently (i.e. to reach the target with a short path length). The task was made more intricate by allowing for nonlinear acceleration of the boat. The speed in a single cardinal direction of the boat was proportional to $0.667t^3 - t^2 + 1.82 * 10^{-14}t$, where t was the amount of time a single cardinal action was selected. t was chosen to be a continuous, rather than discrete variable, and a single action selection resulted in a value of t equal

to the inter-frame interval of the GUI software used. During timesteps where the null action was selected, t would decay in chunks equal to the inter-frame interval. Because of the non-linearity of acceleration, it was possible to move to the target dock with much more efficient gas consumption by using actions in large consecutive presses, rather than by "pulsing" actions (i.e. repeatedly pressing and releasing a single action or switching between cardinal directions). Successful subjects learned to compromise between short path length and minimal gas consumption during the task.

To generate a dataset for our RL agent to similarly learn this compromise, we define our multivariate reward space as the vector space over two dimensional vectors. Each vector contains the negative distance to the target as well as the current gas level. The negative distance to the target was used as a reward to motivate the artificial agent to make progress towards the goal and the gas level was used to efficient progress. In addition, we define the state space of the system to be the two dimensional coordinates of the boat at a given time point. With this information, we are able to compile the human behavior into a dataset of optimal state-action-reward trajectories that can be used by our RL agent.

5.2.2 Multivariate Reward Dynamics Can be Effectively Modeled

In Figure 5.2 we present a summary of the paths taken through the two-dimensional space of rewards. This histogram was taken from a characteristic subject, and the performance demonstrates a general trend from some position on the horizontal axis to some position on the vertical axis over the course of a trial. With respect to the planning game, this trend corresponds to initially starting the game a random distance from the dock, then

over the course of a trial, moving towards the dock while consuming gas. The overall performance in the game is based on the ability to make progress towards the dock while using as little gas as possible. This is accomplished by taking advantage of the nonlinear acceleration provided by the experimental setup: the speed increases nonlinearly with the amount of time consecutive actions are taken in the same direction. Subjects learn during the practice session that the most efficient way to move towards the target is to avoid "pulsing" the throttle (i.e. repeatedly changing direction or accelerating in small bursts). Even with this understanding, paths taken over individual trials were highly variable and while Figure 5.2 suggests that these paths might be linear, Figure 5.4 demonstrates that they are not. In Figure 5.4, the blue points are the observed rewards generated from playing the motor planning game. These paths are highly nonlinear, suggesting the need for a nonlinear function approximator for use as the reward model.

We parameterized the reward model $\mathcal{L}(\tau|\xi)$ using a deep neural network regressor. Details regarding the regressor and the model fitting procedure are given in the Supplement. We demonstrate the modeling error over 20 different training sessions in Figure 5.3 (mean curves are shown with error bars corresponding to the standard deviation of the error for that iteration). The model converges to an average error of about 0.4, where the error is defined using Equation 5.6. To demonstrate the quality of this fit, we provide a few example reward trajectories in Figure 5.4 where the model does particularly poorly. The predicted rewards are shown in orange, while the observed rewards are shown in blue. There are some mistakes around non-smooth transitions (e.g. in the plot on the bottom, left), but qualitatively, the model seems to capture the observed path. We present further validation of the model in the next section, where we use it to design a model-based RL agent.

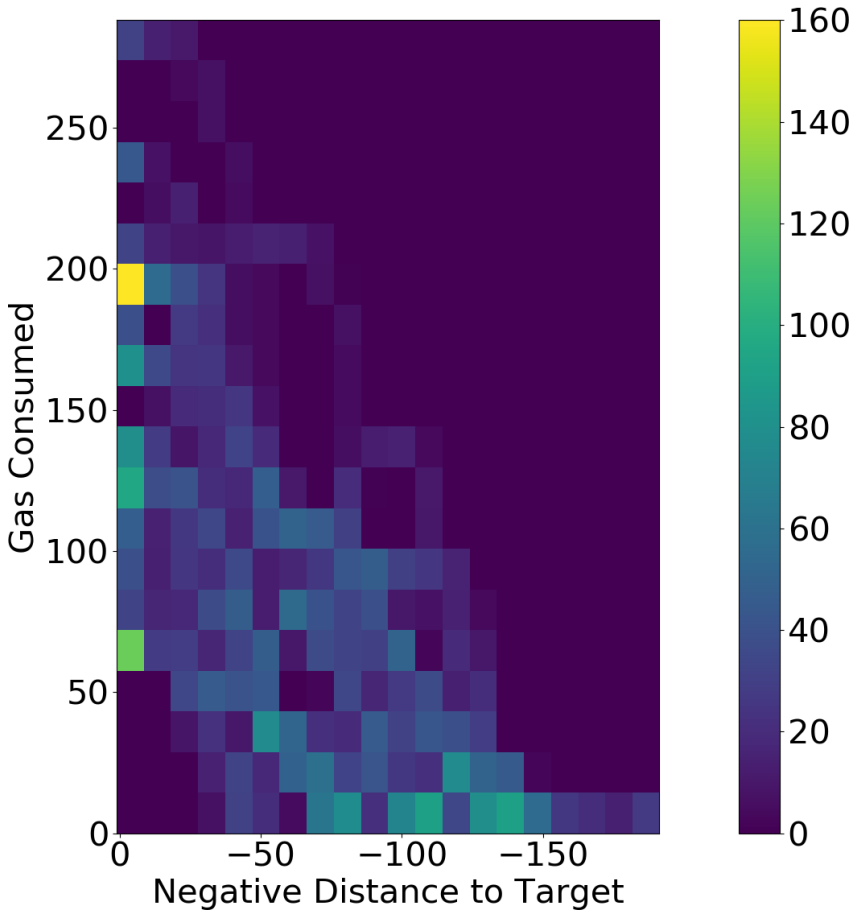


Figure 5.2: Example histogram demonstrating overall performance on the motor planning task from a single subject. This was generated by compiling all trials from a characteristic subject. The general trend is from some position on the horizontal axis to some position on the vertical axis, though paths generated over individual trials were highly variable. This histogram illustrates the progress subjects make in our proposed two-dimensional reward space, consisting of gas level and the negative distance to the target. This suggests that an artificial agent may learn to behave similarly from the subjects behavior in this reward space.

5.2.3 Model-Based Policy Optimization Generates Human-Level Performance

We implemented a controller which makes use of our model of optimal human behavior through the multivariate objective space. Specifically, we used a model-based reinforcement learning framework to perform policy optimization. In this scheme, we use rollouts

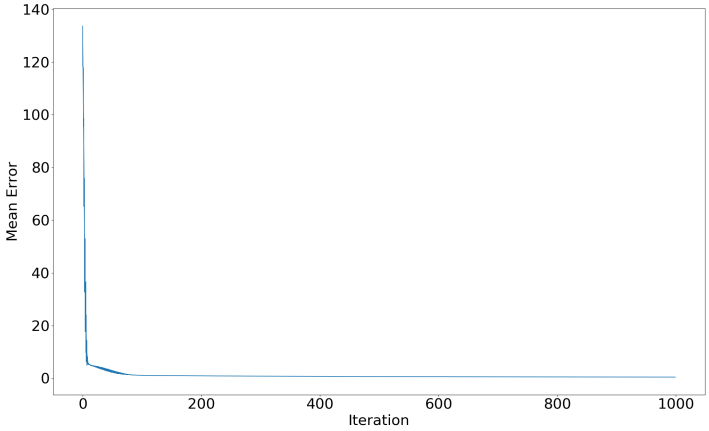


Figure 5.3: The model of optimal reward dynamics converges during training to an average deviation of 0.4. The error is measured by the L2 norm of the difference between the reward predicted by the reward model and the observed reward. The average is taken over a randomly sampled batch of 64 transitions. This plot demonstrates that we are able to find a local optimum when solving Equation 5.6.

forward in time to select actions at the current timestep. This strategy involves simulating forward in time the trajectory resulting from taking a given action at the current timestep and is illustrated in Figure 5.5. Because the action space used consists of four possible actions at time t, we are able to estimate the future cost of taking each possible action, and pick the action that performs best. This strategy is a kind of 0'th order shooting method for policy optimization [112]. The objective for estimating the future cost of an action can be written as

$$\min_{a_t,\ldots,a_T} \sum_{t=t'}^{t'+T} c(r_t^*, r(s_t, a_t)), \text{ s.t. } s_t \sim p(s_t|s_{t-1}, a_t), \tag{5.7}$$

where r_t^* is an optimal multi-variate reward at timestep t, $r(s_t, a_t)$ is the observed multi-variate reward at timestep t, $p(s_{t+1}|s_t, a_t)$ is the model of system dynamics, and $c(\cdot, \cdot)$ is a cost function to be optimized. In this case, $a_t, \ldots, a_T$ are found that approximately solve this optimization problem and a_t is used by the agent at time t. This optimization

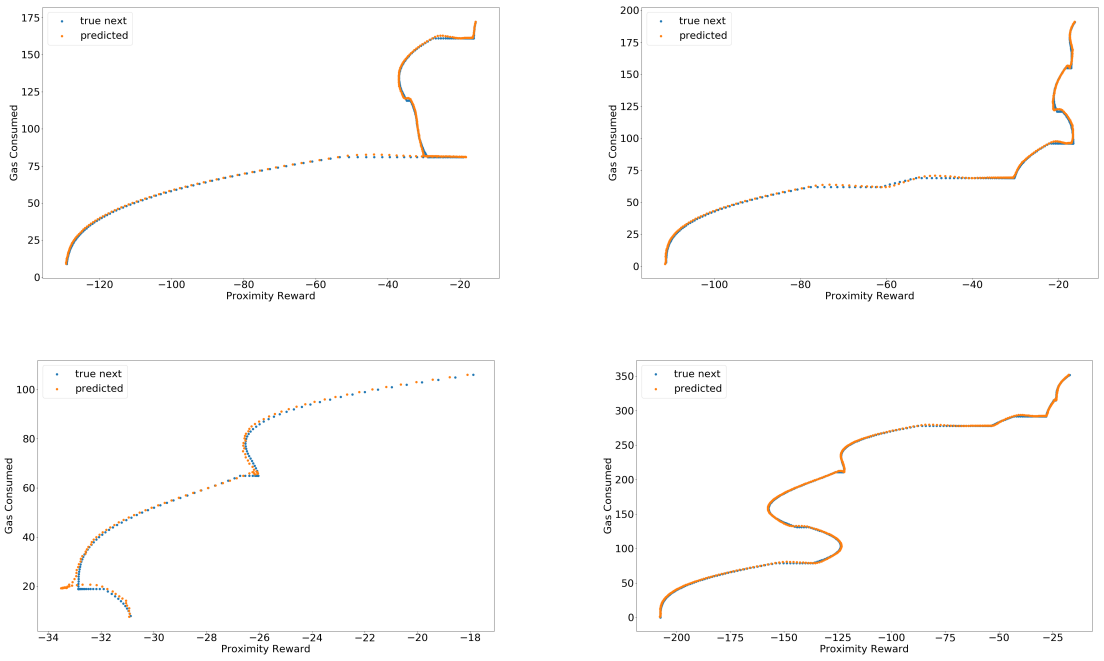


Figure 5.4: Various representative ground truth reward trajectories (blue) and the corresponding predicted trajectories (orange). The largest errors are made by the reward model around non-smooth movements through the reward space. These movements correspoond to drastic changes in direction.

problem is then solved again at the next timestep for $a_{t+1}, \ldots, a_{T+1}$ and a_{t+1} is then used by the agent at time $t+1$. To generate r_t^*, we use the model $\mathcal{L}(\tau|\xi)$, whose performance was described in the previous section. The cost function used is the L_2 norm of the difference between r_t^* and $r(s_t, a_t)$. Similarly, the model of the system dynamics is trained separately to optimize the L_2 norm of the difference between s_{t+1} and s_{t+1}^* where s_{t+1}^* is the actual state at $t+1$ and s_{t+1} is the predicted state at $t+1$.

Example state trajectories generated by the policy are shown in Figure 5.6. It can be seen that the policy produces a small amount of pulsing, which appropriately balances the goals of needing to make progress towards the dock vs. conserving gas. The success of the model of system dynamics as well as the reward model enable this accomplishment.

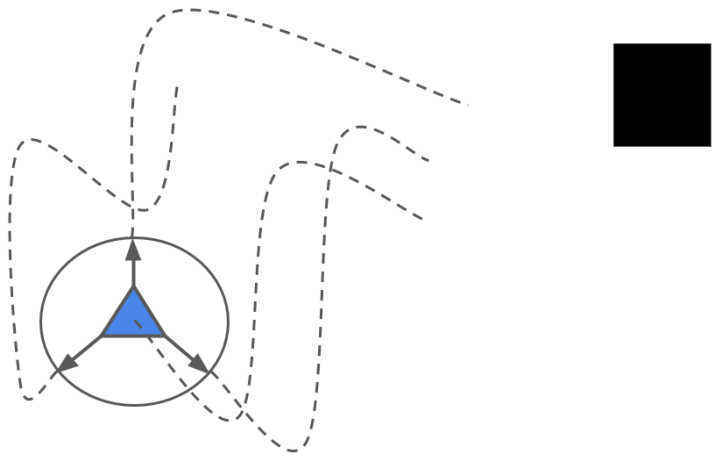


Figure 5.5: Illustration of the policy optimization scheme. Shown is the boat with the four actions at time t superimposed on it and a possible target location (the black square). The solid circle around the boat gives the boundary of the boats state at $t+1$. The dotted lines illustrate simulated rollouts forward in time until time $t+T$, from each possible action at t. These rollouts demonstrate that certain actions at t are superior to others in that they allow for fewer turns (reduced gas consumption) while making better progress towards the target.

The former is demonstrated by the blue curve, which is the predicted state trajectory, while the red curve is the observed state trajectory. Qualitatively, the system dynamics model appears to appropriately characterize the true dynamics. The success of the policy is quantitatively validated in Figure 5.7, where we produce histograms comparing human and machine performance using the remaining gas and path length over all trials. Machine performance is shown in blue, while human performance is shown in orange. These histograms were generated using machine performance on 500 trials, the conditions for which were randomly generated using the same process that was used to generate the human trials. The policy produced using Equation 5.7 results in gas consumption and efficient progress towards the dock comparable to that seen in the human subjects.

The plots shown in Figure 5.7 were generated using successful trials only. The failure rate, 17%, of the policy was slightly higher than that observed by human subjects, where the

worst performing human subject failed about 15% of the trials. This can be attributed to two causes: first, human subjects were at a slight advantage in that they were able to choose between two different boats which had different action spaces. Depending on the location of the dock relative to the starting location, the human subjects were able to choose the boat that would allow them to move towards the dock most efficiently. We gave the policy only one boat, thus it was forced to use a single action space for all trials. Since most of the failures of the artificial policy resulted from running out of gas, having access to a single boat was a significant handicap. A second likely cause of the increased failure rate is the limited dataset of optimal trajectories. Because the reward model is trained on observed, optimal reward trajectories and the initial and dock locations were randomly generated for each trial, it is likely that most of the possible optimal reward paths were not included in the dataset. Adapting the policy to account for this limitation requires a transfer of skill from one domain of expertise to another, something that was not accounted for in our policy optimization scheme. It may be an interesting direction for future work to incorporate transfer learning into our MOIRL method.

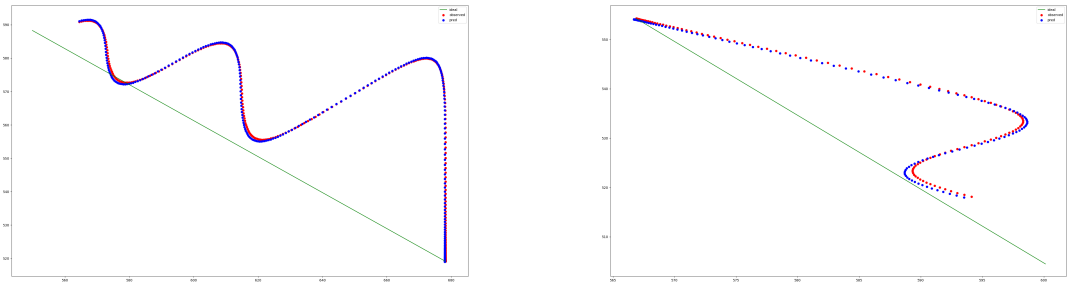


Figure 5.6: Example paths through the state space generated by the MOIRL controller. The green line is the shortest possible path between initial and target locations, the blue curve is the path predicted by the model of the system dynamics, and the red curve is the observed path. It can be seen that the policy generates a small amount of pulsing and as a result, balances the objectives of needing to take the shortest possible path while conserving fuel.

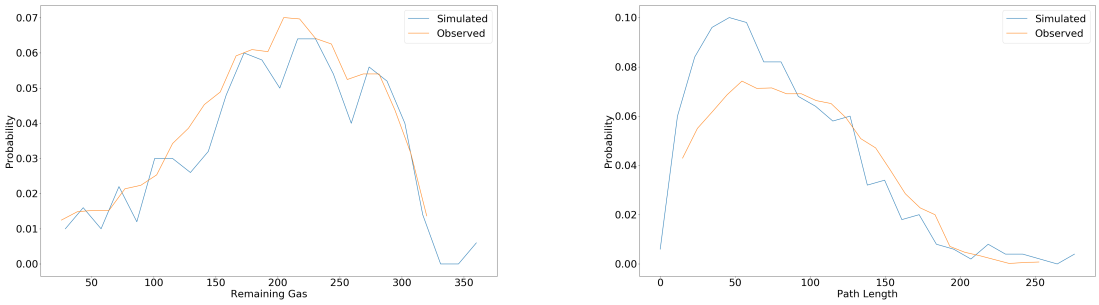


Figure 5.7: Performance of RL agent compared with human subjects. The performance of the artificial agent is comparable to the human subjects on both gas consumption and path length.

Chapter 6

Network Minimum Free Energy Learning

Human learning can be naturally described as an optimization problem. Many past models of learning have defined a performance objective and shown that the brain encodes this objective and uses it to assess its own performance. For example, in the case of tasks related to motor control, neural activity is often decomposed into regions encoding a model of the actuators and regions encoding the relationship between those actuators and the reward for successful task performance [80, 83, 84, 81, 82]. Learning objectives can be formulated in this regime using state-prediction and reward-prediction errors, respectively. The model of learning that results is based upon an assumption that modeling of motor actuation and reward anticipation errors are segregated in the brain in some manner.

A recent development in computational neuroscience allows us to model learning while avoiding this assumption. The brain has been shown in a number of previous results to optimize a free energy functional during learning [85, 70, 63, 67, 66, 69, 68, 64, 62]. We

extend the use of free energy functionals to model human learning by defining a functional over a network. This functional extends previous models applied to neural systems in a number of ways [85]. First, it allows for the activity of a potentially large number of brain regions to be explicitly modeled (we use 138 in this work). Next, it allows for the modeling of the learning objectives of each brain region, independent of every other. Since the brain regions considered in this work consist of many thousands of neurons with their own local dynamics, it is essential that region specific objectives are used to characterize the learning that takes place within brain regions, as well as across the population of regions. Finally, interactions between brain regions over the network are also explicitly modeled (via an *interaction matrix*). The interpretation of this interaction matrix is analogous to that of functional connectivity matrices [72, 73, 74, 75, 61]. It is different from past work on functional connectivity in that we don't explicitly define an interaction function (eg., correlation or coherence). Instead, it is entirely inferred from data.

We refer to the optimization of this functional as the Network Minimum Free Energy Learning (NMFEL) model and show that this model can characterize human learning. We accomplish this by first showing that a particular optimization scheme (a Wasserstein Gradient Flow) over this functional generates a Fokker-Planck equation over a network (NFPE). We fit this equation to a timeseries of fMRI BOLD activity generated during the learning of a discrete sequence production (DSP) task. In doing so, we are able to observe that human learning, from naive to expert ability, is nonlinear and requires an NFPE with time-varying parameters. This result is explained, intuitively, by the fact that humans utilize a number of distinct objectives during the course of learning a complex motor sequencing task.

The NFPE is a model of global neural dynamics, and we show that it can be used to derive global changes from the activity of individual brain regions. We accomplish this using a sensitivity analysis applied to the NFPE. The output of this analysis is a set of sensitivities for each brain region: these indicate how "sensitive" the full trajectory of neural activity is to perturbations of individual brain regions. The results of this analysis show that global neural dynamics are most strongly perturbed by the sensory, motor, and visuospatial processing regions long known to be associated with motor learning.

With this method for explaining global dynamics in terms of local fluctuations, we show how the cumulative performance of individual subjects over time can be explained using isolated brain regions. We correlate regional sensitivities of individual subjects with their respective cumulative performance. We find, in contrast to the first sensitivity analysis, that the sensitivities of a number of brain regions thought to participate in higher level cognition and meta-learning (prefrontal cortices and basal ganglia) are strongly correlated with cumulative performance.

The DSP task requires subjects to simultaneously learn the association between a symbolic cue and a 10 element motor sequence. Initially, subjects rely on visual guidance to complete individual movements as they acquire knowledge of the different sequences. After 200-400 trials, subjects have explicit knowledge of each sequence and can draw on verbal overshadowing to internally guide their sequential movements. After several thousand practice trials, the sequences are executed faster than verbal overshadowing or other conscious strategies can sustain, suggesting that sequential memories have been learned within motor actuation circuits. Thus, subjects are transitioning through different objectives during learning and their strategies in what objectives to adopt be thought of as a form of meta-learning [113]. We hypothesize that variable activity of this meta-

learning network is capable of inducing different learning algorithms. We validate this idea through a simulation study, where we show that perturbation of this meta-learning network is capable of inducing a distribution over paths indistinguishable from that generated by the population of subjects. This lends support to the idea that optimization of the activity of this meta-learning network can select from amongst a set of learning algorithms.

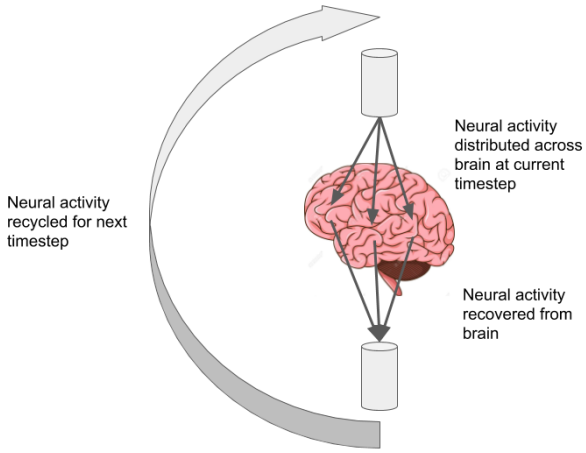


Figure 6.1: Cartoon of NFPE model. Because neural activity is represented as a probability vector, the total amount of neural activity is fixed (equal to 1 in this chapter). Neural dynamics then involve distributing this activity in a time-varying fashion. At each timestep, activity is distributed, collected, and recycled to be redistributed at the next timestep. The manner in which activity is distributed depends on the objectives of the individual brain regions, their pairwise interactions, and smoothing the activity over the brain by optimizing the entropy of the total distribution.

6.1 Data Collection

In our longitudinal study of motor learning, participants (N=20) performed a discrete sequence production task for six weeks. We varied the amount of practice across a set of six sequences [114]. Neural responses (BOLD activity) were recorded to obtain baseline neural responses while the participants were first being exposed to the sequences (INIT). Participants then completed at-home training sessions where two of the sequences

were trained extensively (64 trials/session; EXT), two were trained moderately (10 trials/session; MOD), and two minimally (1 trial/session; MIN). Performance was measured by movement time to complete the sequence. Across the four imaging sessions (with two weeks separation between each), whole-brain analysis was conducted to identify brain regions activated during sequence production for each training intensity condition. We denote these measurements as COND1, COND2, and COND3 (e.g. MIN1, MIN2, and MIN3 for the MIN condition). Beta values (derived from modeling BOLD activity using a GLM) from whole brain analysis were extracted using the Harvard-Oxford atlas.

6.2 Network Minimum Free Energy Learning

We represent neural activity as a probability vector, $p = [p_1, \ldots, p_B]$, where B is the number of brain regions. The are two reasons for this: first, it allows us to extend existing results characterizing neural activity as a probability density [85]. In this past work [85], global neural activity was modeled as a single random variable. In contrast, treating neural activity as a probability vector allows us to directly model the activity of each brain region. Next, since $\sum_i p_i = 1$, it allows us to fix global neural activity as constant. Because of this, we are able to model relative neural activities. This is advantageous in our experimental setting: data was collected across a diverse population of subjects and times, and absolute neural activities can vary significantly in response to stimuli outside the experimental protocol. Moreover, it has been shown that absolute neural activity can vary with training intensity [85]. It was essential for us to remove this effect to be able to compare results across MIN, MOD, and EXT conditions.

We then define the free energy functional for p to be

$$\mathcal{F}(p) = \mathrm{E}_p[V] + \mathrm{E}_{p\times p}[W] - \mathrm{H}[p],$$

where $\mathrm{E}[\cdot]$ is the expectation operator, V is a potential for individual brain regions, W is an interaction potential between brain regions, and $\mathrm{H}[\cdot]$ is the entropy. Because p is a probability vector, $\mathcal{F}$ expands into

$$\mathcal{F}_{NMFEL}(p) = \sum_i V_i p_i + \sum_i \sum_j W_{i,j} p_{i,j} + \sum_i p_i \log(p_i). \tag{6.1}$$

Incorporation of $\mathcal{F}_{NMFEL}$ into an optimization problem yields the following model

$$\min_p \mathcal{F}_{NMFEL}(p), \tag{6.2}$$

which we refer to as the Network Minimum Free Energy Learning (NMFEL) model. The term $\sum_i V_i p_i$ quantifies the objective optimized by individual brain regions, independent of all others. Since the brain regions we consider in this work are quite complex and consist of many thousands of cells, they are also adapting to training. This effect must be included in the model. Second, the term $\sum_i \sum_j W_{i,j} p_{i,j}$ quantifies the interactions between brain regions that is optimized during learning. It has been repeatedly demonstrated that interactions between brain regions over a network change as a result of learning [72, 74, 75, 61]. These approaches often require *a priori* knowledge of an interaction function (e.g. the Pearson correlation has been used). The NMFEL objective allows us to define a generic quadratic interaction potential and then recover its values in a data-driven manner by fitting the NMFEL to neural activity. Finally, the entropy term, $\sum_i p_i \log(p_i)$, results in smoothing of the neural activity during optimization. This smoothing prevents the accumulation of probability mass by a small number of brain regions during optimization of the NMFEL in simulation, an effect which is unlikely in

healthy human beings. With the NMFEL objective in Equation 6.2, we are able to derive a model of neural dynamics. By taking a 2-Wasserstein Gradient flow over Equation 6.1 (derivation shown in the Supplement), we are able to retrieve a Fokker-Planck equation over a network. The dynamics for p_i are given by

$$\frac{dp_i}{dt} = \sum_{j \in N(i)} \omega_{i,j}\theta_{i,j}(V_j - V_i + (Wp)_j - (Wp)_i)$$

$$+ \sum_{j \in N(i)} \omega_{i,j}\theta_{i,j}(\log(p_j) - \log(p_i)), \qquad (6.3)$$

where $N(i)$ is the set of neighbors of the i'th node, $\omega_{i,j}$ is the edge-weight between the i'th and j'th nodes, and $\theta_{i,j} = \frac{p_i + p_j}{2}$ defines the flux of probability mass over an edge. In this work, we assume the brain regions lie on a fully connected network with uniform edge weights, though extensions to incorporate estimates of the structural connectivity network are possible. We refer to Equation 3 as the Network Fokker-Planck equation (NFPE). For the subsequent results, we directly solve the discretized version of the NFPE in Equation 3 by solving the set of coupled ODE's. A cartoon illustrating the NFPE is shown in Figure 6.1, and details about its solution are given in the Supplement.

6.3 NMFEL Learning Paths

We validate the NMFEL model by fitting its parameters (V and W) to the fMRI data from the sequence production task. We take advantage of the fact that the Gibb's distribution is the steady-state of the NFPE and is given by

$$p_i = \frac{1}{Z} e^{V_i + (Wp)_i}.$$

We treat the final time evolution of each condition as the steady-state for that condition (MIN3, MOD3, EXT3) and fit V and W for each condition using these distribu-

tions. These parameters were optimized using a variant of stochastic gradient descent over the average KL-Divergence between the estimated distribution and the probability vectors generated from the population of subjects (details of these fits are given in the Supplement). The resultant parameters were used to simulate the NFPE forward in time to approximate the average probability vector for each condition: this process resulted in small deviations in the total variation norm (MIN3: TVD(0.00118); MOD3: TVD(0.00132); EXT3: TVD(0.00117)).

Ideally, in simulating the NFPE forward in time from the initial condition, the probability vector would pass through COND1 and COND2. But we found that this was not possible, as forward simulation of the NFPE would yield a path close to a straight line between INIT and COND3. This issue is highlighted in Figure 6.3. For each condition, we found that learning does not progress along a straight line, that is, COND1 and COND2 do not lie on the line between INIT and COND3. In fact, the total distance traversed during each condition (estimated using the total variation norm) is much greater than a straight line (MIN: Straight(0.00476) vs Observed(0.00996); MOD: Straight(0.00417) vs Observed(0.01118); EXT: Straight(0.00392) vs Observed(0.01499)).

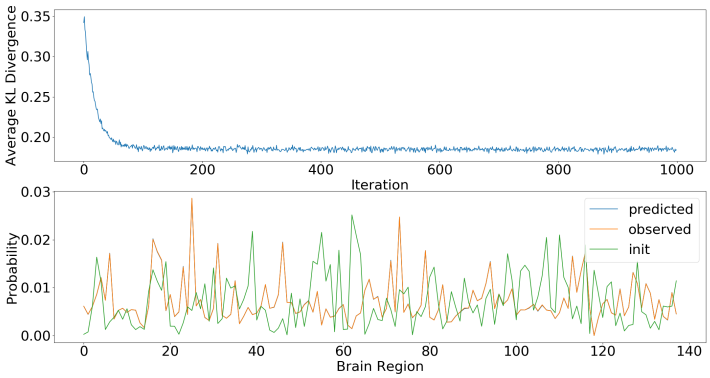


Figure 6.2: Example output from parameter estimation characteristic of overall performance. **Top**: optimization objective against iteration number. The optimization appears to converge after about 100 iterations. **Bottom**: comparison of the fitted (blue) with the observed distribution (orange) with the initial distribution given as a reference (green). The predicted and observed distributions are indistinguishable without increasing the resolution of the vertical axis by an order of magnitude.

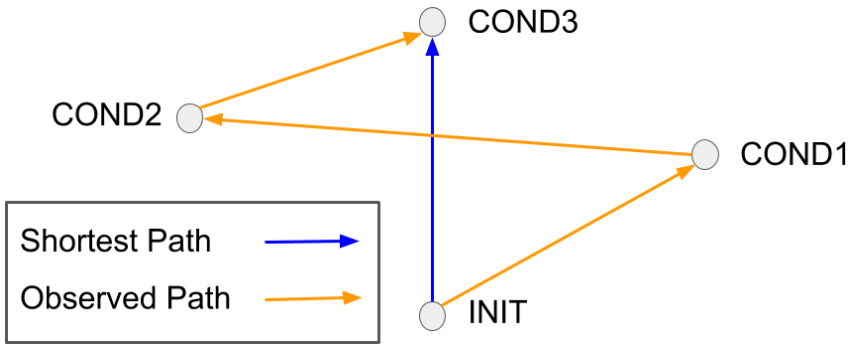


Figure 6.3: Illustration of the observed learning path compared to the shortest path. The shape of the observed paths don't necessarily correspond to that in the illustration, the figure is meant to illustrate the difference between the observed and linear paths. For all conditions, the observed paths are much longer than the linear path.

Distances Traversed During Learning			
Condition	MIN	MOD	EXT
INIT → COND1	0.00304	0.00349	0.00539
COND1 → COND2	0.00428	0.00361	0.00496
COND2 → COND3	0.00264	0.00408	0.00464
INIT → COND3	0.00477	0.00417	0.00392

Table 6.1: Distances traversed between time points for each condition. The metric used was the total variation norm. For all conditions, the INIT → COND3 distance was not observed.

One problem raised by the nonlinearity of learning paths is that the human brain isn't optimizing a set of time-invariant parameters of the NFPE model. From a psychological perspective, the implication of this point is that there exist intermediate objectives optimized by the brain at different points in the learning process. To capture this behavior, a slightly more sophisticated, time-varying NFPE model is required. To this point, we fit Gibb's distributions to each run (COND1, COND2, and COND3). We then simulated the NFPE forward in time, switching the parameters after a time interval of fixed length. In this way, we were able to obtain high-quality fits (estimated using the total variation norm) for the entire timeseries of neural activity, for each condition (MIN: 1(0.00184), 2(0.00197), 3(0.00118); MOD: 1(0.00087), 2(0.00160), 3(0.00131); EXT: 1(0.00147), 2(0.00155), 3(0.00116)).

MIN Condition	
Brain Region	Sensitivity
L Lingual Gyrus	-34.26
R Intracalcarine Cortex	-27.50
L Superior Parietal Lobule	-25.37
L Intracalcarine Cortex	-25.06
R Lingual Gyrus	-24.82
L Precentral Gyrus	-24.71
L Cuneal Cortex	-24.51
L Postcentral Gyrus	-24.21
R Cuneal Cortex	-23.76
R Supracalcarine Cortex	-22.99
LR CB Vermis X	21.32
MOD Condition	
Brain Region Name	Sensitivity
L Supracalcarine Cortex	-17.96
L Cuneal Cortex	-17.70
L Intracalcarine Cortex	-15.81
R Supracalcarine Cortex	-15.61
L Lingual Gyrus	-15.56
R Cuneal Cortex	-14.72
R Lingual Gyrus	-12.82
L Postcentral Gyrus	-12.52
LR CB Vermis X	11.51
R Intracalcarine Cortex	-11.25
L Superior Parietal Lobule	-10.82
EXT Condition	
Brain Region Name	Sensitivity
R Intracalcarine Cortex	-9.45
L Intracalcarine Cortex	-8.88
L Lingual Gyrus	-8.70
L Postcentral Gyrus	-8.06
R Supracalcarine Cortex	-7.05
R Planum Polare	6.79
R Lingual Gyrus	-6.79
L Superior Parietal Lobule	-6.36
LR CB Vermis X	6.01
L Supramarginal Gyrus	-5.64
R Cuneal Cortex	-5.58

Table 6.2: Output of the sensitivity analysis on population average global dynamics. The regions that are returned are known to participate in processing of sensation (tactile and visual) with motor control.

6.4 Local Influence on Global Dynamics

The results discussed so far have concerned global neural dynamics and do not clarify how these global dynamics relate to individual brain regions known to participate in motor sequence learning. To address this issue, we perform a sensitivity analysis of the NFPE dynamics to isolate individual brain regions that contribute strongly to the evolution of our model of learning. To accomplish this, we take advantage of recent work involving the automatic differentiation of continuous time neural networks [115]. We note that the discretized NFPE presented in Equation 6.3 3 is a set of neural ordinary differential equations, and we define a loss function as the L2 norm of the difference between the simulated neural activity at time t and the neural activity collected at the end of the current phase (if the current phase will converge on COND2, we compare the simulated neural activity with COND2). The neural activity used was the mean probability vector taken over the entire population of subjects. We then ran an adjoint solver to backpropagate this loss from the final time point to the initial time point, yielding a gradient of the loss with respect to the neural activity at the initial time point. This gradient gives us a measure of the influence of each brain region on the overall learning process (i.e. the paths taken by global neural activity). Details on the method used are given in the Supplement.

We present a summary of these results in Table 6.2, where the ten brain regions with highest magnitude sensitivities are shown for each condition. The output of this analysis includes regions generally known to participate in motor sequence learning. In particular, global dynamics are sensitive to regions involved in sensory reception and motor execution (postcentral and precentral gyri, SMA) and the coordination of sensory reception with motor activity (superior parietal lobule, supramarginal gyrus, SMA, parietal operculum). The signs of the sensitivities are particularly relevant, in addition to their

magnitudes. For example, the cerebellar sensitivities are positive, indicating that increased cerebellar activity leads to increased deviation of the simulated dynamics from the observed dynamics. This is consistent with the understanding that the cerebellum provides negative feedback during motor control: increased cerebellar activity indicates the presence of errors in task execution. Large positive sensitivities can also be used to identify regions that are not important for the task. The large positive sensitivity of the planum polare in the EXT condition indicates that activation of this region is likely counterproductive during task execution.

6.5 Learning Algorithm Selection

Sensitivity analysis can also be applied to the neural activity trajectories of individual subjects. In particular, we explore the relationship between the sensitivities that produce individual trajectories with the associated cumulative performance over time on the task. To accomplish this, we define a loss function as the L2 norm of the difference between the simulated neural activity at time t and the neural activity for a given subject on the current phase of the learning process. The output of our adjoint method is then a collection of sensitivities that indicate how the population mean model might be perturbed to yield the observed neural activity for a given subject. We then correlate the sensitivities for each brain region with the cumulative performance generated by the population of subjects. The performance measure used was the function $\gamma^3 m_I + \gamma^2 m_1 + \gamma m_2 + m_3$, where m_I is the average movement time on the INIT condition, m_1 was the average movement time on COND1, m_2 was the average movement time on COND2, m_3 was the average movement time on COND3, and $\gamma = 1.5$ was used. γ was selected to identify regions involved in selecting the paths taken by individual subjects, where early performance is more important for this process than later performance. We used the Pearson correlation

test with the null hypothesis that the sensitivities and performances are uncorrelated. The results of this experiment are shown in Table 6.3, where a p-value threshold of 0.05 was applied. Outlier removal was performed on the sensitivities using a fixed threshold of twice the standard deviation of the sensitivities. Example plots of cumulative performance against the sensitivities are shown in figure 6.4 to demonstrate the effect of this processing.

Pearson Correlation Between Cumulative Performance and Sensitivity		
Brain Region	Correlation	p-value
R Amygala	-0.73	0.0003
L Frontal Medial Cortex	-0.64	0.0042
R Occipital Fusiform	0.62	0.0043
R Superior Parietal	-0.60	0.0050
R Precentral Gyrus	-0.62	0.0059
R Lingual Gyrus	0.62	0.0077
R Putamen	-0.57	0.0106
R Frontal Medial Cortex	-0.57	0.0110
L Inferior Temporal Gyrus	0.602	0.0134
L Frontal Operculum	-0.532	0.0189
R Heschl's Gyrus	0.519	0.0227
LR CB Vermis VIIb	-0.548	0.0229
L Lingual Gyrus	0.53	0.0236
L Subcallosal Cortex	-0.51	0.0256
R Angular Gyrus	0.50	0.0345
R Superior Temporal Gyrus	0.494	0.0373
R Cuneal Cortex	0.484	0.0490
R Pallidum	-0.469	0.0499

Table 6.3: Results of Pearson Correlation Test between regional sensitivities and cumulative task performance. Along with a collection of sensorimotor regions known to participate in motor learning, we observe members of the pre-frontal cortex and basal ganglia with high–magnitude correlations.

To facilitate interpretation of the results, we restrict this sensitivity analysis to the EXT condition. This choice was made because the performance of most subjects in the MIN and MOD conditions hadn't plateaued by the end of the study, thus comparisons of per-

formance across conditions would be of questionable value. This analysis again uncovers a number of regions known to participate in motor control (precentral gyrus, lingual gyrus, cerebellum) and visuospatial processing (cuneal cortex, inferior temporal gyrus, superior parietal lobule) as well as regions likely uninvolved in motor sequence learning (Heschl's Gyrus). Unlike the earlier sensitivity analysis, a number of regions involved in reward circuitry also emerge (putamen, pallidum, amygdala, frontal medial cortices). These regions appear to be encoding cumulative performance based on their influence on the paths taken by global neural activity (e.g. subjects that perform worse on the task tend to have higher activity of these regions).

This result is related to recent theories that suggest that a Prefrontal Cortex/Basal Galgia (PFC/BG) circuit is capable of selecting learning algorithms to tune for optimal performance or adjust for a non-stationary task reward (meta-learning or learning-to-learn) [116]. Motor sequence learning tasks can draw on multiple strategies to enhance performance and the method for selecting amongst these strategies can be treated as a form of meta-learning [113, 117, 118, 119]. With that in mind the population of subjects induces a distribution over learning paths, each of which may be considered a distinct learning algorithm (i.e. an instance of the NMFEL parameterized by a distinct set of parameters). We hypothesize that the PFC/BG circuit is capable of generating this distribution. To validate this hypothesis, we perform a simulated experiment where we perturb the parameters of the NMFEL associated with the PFC/BG regions recovered by the sensitivity analysis above (putamen, pallidum, frontal medial cortices) to demonstrate the ability of these regions to induce a distribution over learning paths indistinguishable from the one observed in our population of subjects. We accomplish this by fitting a distribution over these parameters (a multivariate normal with diagonal covariance) using maximum likelihood estimation. The likelihood function used was a normal distribution

over path lengths, fit to the set of path lengths generated by all subjects. We generated candidate batches of path lengths by repeatedly sampling sets of NMFEL parameters and forward simulating the NFPE with these parameters. With this procedure and using a 2-sample Kolmogorov-Smirnov test, we were able to generate sets of path lengths statistically indistinguishable from the observed path lengths for each condition (MIN: KS=0.1, p>0.5; MOD: KS=0.2, p>0.5; EXT: KS=0.199, p>0.5). This result lends support to the idea that the PFC/BG circuit is capable of inducing meta-learning by showing that variable activity of these regions is capable of inducing a set of learning paths consistent with that observed in human learning.

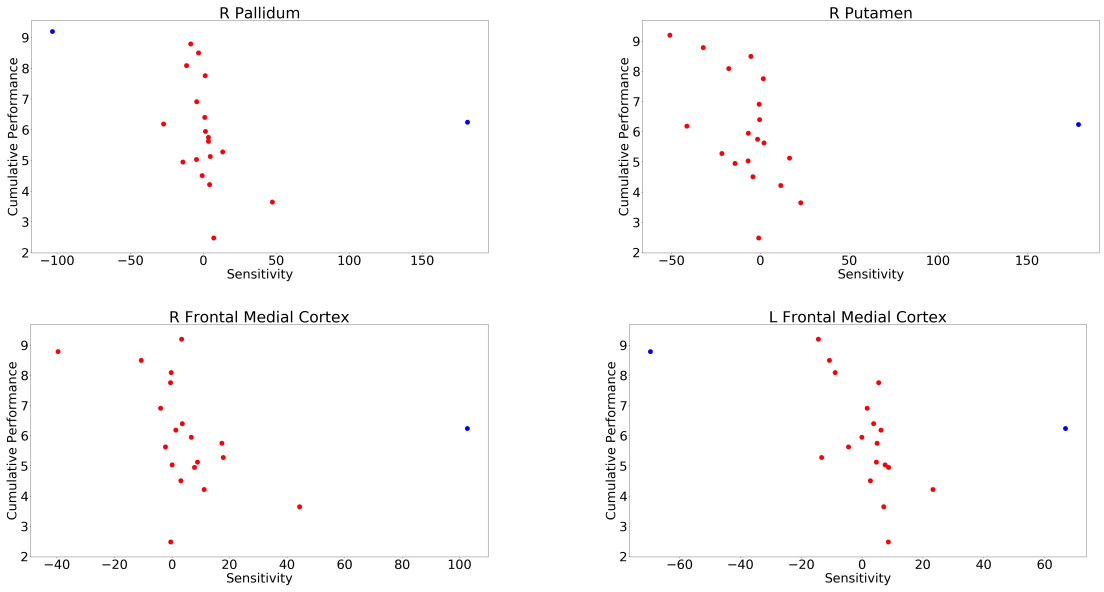


Figure 6.4: Correlation analysis between EXT sensitivities of individual subjects and their cumulative performance on the task. Regions involved in reward circuitry (frontal medial cortices, basal ganglia, amygdala) are strongly correlated with cumulative performance. These correlations are illustrated in the four plots shown, where each point is a subject. Red points are included in the analysis and blue points are excluded as outliers. These plots demonstrate strong linear trends.

6.6 Discussion

We have shown in this work how the minimum free energy framework can be extended to model systems supported on a network. This model allows the user to accurately model global neural dynamics during learning as well as to derive those dynamics based upon the activity of individual brain regions using sensitivity analysis. We have further shown that sensitivity analysis can be used to illustrate how the variable performance of individual subjects is encoded in the paths taken by their global neural activity. The distribution over learning paths induced by the population of subjects was explained using the theory of meta-learning. These results lend support to the idea of minimum free energy learning acting as a unifying principle of neural learning in that the assumptions we have made about the goals of human learning are far weaker than what is commonly used.

The use of free energy functionals as a model of the learning objective has a number of advantages over past approaches. First, it allows us to explicitly quantify the uncertainty in neural activity (e.g. as generated by a population of human subjects) by defining a distribution over neural activity. Incorporation of Bayesian methods of validation is much more straightforward in this setting than in non-probabilistic ones. We have shown how variable PFC/BG activity can induce uncertainty in global neural trajectories. A more general approach to uncertainty quantification is possible, for example one that attempts to partition the uncertainty in the learning path chosen based on contributions from individual brain regions. Next, the free energy functional may have a number of different interpretations, including but not limited to, state and reward prediction. For example, a number of recent models of learning include objectives that do not fit neatly into either state-prediction or reward-prediction categories: for example, competitive environments that require prediction of an opponent's behavior require error representations that are

arguably difficult to categorize in this way [120, 121, 122]. Free energy functionals can be seen as a generalization of these more specific reward objectives.

Further exploration of the relationship between NMFEL and meta-learning is also needed. Meta-RL is thought to arise as an emergent property of the fact that the PFC exists in a recurrent circuit that includes basal ganglia and thalamic regions. Because the PFC itself is capable of RL, model-free tuning of PFC circuitry by the basal ganglia is postulated to give rise to novel RL algorithms: the complete algorithm is referred to as Meta-RL. The NFPE is itself a kind of RL algorithm and is often used in a reduced form for RL methods in engineering (without the interaction potential, the NMFEL corresponds to Maximum Entropy RL) [60]. In an engineering setting, meta-learning takes a number of different forms, from intelligently selecting method initializations to fitting optimization algorithms to fitting optimization objectives [18]. It is unclear if the PFC/BG circuit is capable of doing any of these, or even if the categorization produced by the engineering literature is appropriate for understanding neural systems. Further exploration of these issues is necessary to better understand the mechanism of meta-learning in the human brain.

Chapter 7

Conclusion and Future Directions

In this thesis, we've described a number of ways reinforcement learning can be used to both control and better understand large scale neural systems and human behavior. Since all of these works were produced with a view towards eventually applying reinforcement learning to living neural systems, we'll conclude this thesis with a discussion of the current limitations of the works discussed and a few ideas about how to overcome them.

The first and perhaps most concerning limitation of reinforcement learning relates to its safety. Classically, reinforcement learning methods are split into model-free and model-based methods, where the former generally require more experimentation with the system to be controlled. One of the major considerations imposed on the user in selecting one of these approaches involves managing the trade-off between the work required to fit an accurate model and the need for safe exploration during policy optimization. We've explored a unifying approach based on Temporal Difference Models: the Distributional Temporal Difference Model, which is compatible with both model-free and model-based reinforcement learning. It may be an interesting future direction to explore the extent to which the trade-off between model-free and model-based reinforcement learning can

be learned, rather than manually imposed on the algorithm. The approach used to accomplish this might be framed as a multi-objective optimization problem, where two of the objectives are the safety of exploration and the performance of model-fitting. The tradeoff between these two considerations could either be fit online or learned from an agent known to be expert at achieving a good balance.

A completely different approach to safely fitting useful policies using reinforcement learning could be achieved with insight into the transfer of policies across related systems. Fitting policies using reinforcement learning on laboratory animals, even methods which require unsafe exploration, is much more realistic given current methods than doing so on humans. But currently doing so would be unlikely to be particularly useful in generating policies that work on humans because of the poor understanding concerning the transfer of policies. Unlike computer vision and natural language processing, fields where transfer of deep neural networks have resulted in enormous advances, transfer learning for reinforcement learning algorithms is still in its infancy [123, 124, 125]. The ability to fit a policy in a low-risk setting and update it quickly and safely to a higher-risk setting would be extremely valuable and potentially allow for more rapid adoption of reinforcement learning algorithms for controlling biological systems.

Even in the case where a reinforcement learning algorithm could be shown to be performant and safe in controlling a large-scale neural system, the black-box nature of deep neural network function approximators would likely limit their use. The explanability and interpretability of deep neural networks is of critical importance in producing widespread adoption and use. Neural Ordinary and Partial Differential Equations present a path forward in addressing these issues because of the vast literature on ordinary and partial differential equations. The ability to derive a neural network from either an ODE or a

PDE gives the user a valuable set of tools that can be used to justify predictions made by the network in a quantitative, theoretically sound manner. Combined with the ability to fit reinforcement learning algorithms in a safe manner, having explainable and interpretable neural networks will significantly increase the confidence users have in applying them towards systems that are as sensitive as our brains.

Bibliography

[1] E. Brown, J. Moehlis, and P. Holmes, *On the phase reduction and response dynamics of neural oscillators populations.*, *Neural computation* (2003).

[2] P. Danzl, J. Hespanha, and J. Moehlis, *Event-based minimum-time control of oscillatory neuron models: phase randomization, maximal spike rate increase, and desynchronization.*, *Biological Cybernetics* (2009).

[3] G. Orosz, J. Moehlis, and R. Murray, *Controlling biological networks by time-delayed signals.*, *Philosophical Transactions of the Royal Society* (2010).

[4] J. Moehlis, E. Shea-Brown, and H. Rabitz, *Optimal inputs for phase models of spiking neurons.*, *Journal for Computational Nonlinear Dynamics* (2005).

[5] J. Snyder, A. Zlotnik, and A. Hagberg, *Stability of entrainment of a continuum of coupled oscillators.*, *Chaos* (2017).

[6] A. Zlotnik, R. Nagao, I. Kiss, and J. Li, *Phase-selective entrainment of nonlinear oscillator ensembles.*, *Nature Communications* (2016).

[7] S. Shirasaka, N. Watanabe, Y. Kawamura, and H. Nakao, *Optimizing stability of mutual synchronization between a pair of limit-cycle oscilllators with weak cross coupling.*, *arXiv* (2017).

[8] H. Nakao, S. Yasui, M. Ota, K. Arai, and Y. Kawamura, *Phase reduction and synchronization of a network of coupled dynamical elements exhibiting collective oscillations.*, *arXiv* (2017).

[9] A. Nandi, M. Kafashan, and S. Ching, *Controlling point process generalized linear models of neural spiking.*, *American Control Conference* (2016).

[10] A. Nandi, M. Kafashan, and S. Ching, *Control analysis and design for statistical models of spiking networks.*, *IEEE Transactions on Control of Network Systems* (2016).

[11] J. Pineau, A. Guez, R. Vincent, G. Panuccio, and M. Avoli, *Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach.*, *International Journal of Neural Systems* (2009).

[12] G. Panuccio, A. Guez, R. Vincent, M. Avoli, and J. Pineau, *Adaptive control of epileptiform excitability in an in-vitro model of limbic seizures.*, *Experimental Neurology* (2016).

[13] M. Saleh, K. Takahashi, Y. Amit, and N. Hatsopoulos, *Encoding of coordinated grasp trajectories in primary motor cortex.*, *The Journal of Neuroscience* (2010).

[14] J. Chapin, K. Moxon, R. Markowitz, and M. Nicolelis, *Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex.*, *Nature Neuroscience* (1999).

[15] A. Radulescu and L. Mujica-Parodi, *A principal component network analysis of prefrontal-limbic fmri time series in schizophrenia patients and healthy controls.*, *Psychiatry Research* (2009).

[16] S. Pequito, A. Ashourvan, D. Bassett, B. Litt, and G. Pappas, *Spectral control of cortical activity.*, *American Control Conference* (2017).

[17] C. Finn, P. Abbeel, and S. Levine, *Model-agnostic meta-learning for fast adaptation of deep networks.*, *arXiv* (2017).

[18] A. Nagabandi, I. Clavera, S. Liu, R. Fearing, P. Abbeel, S. Levine, and C. Finn, *Learning to adapt in dynamic real-world environments through meta-reinforcement learning.*, *arXiv* (2019).

[19] A. Zhou, E. Jang, D. Kappler, A. Herzog, M. Khansari, P. Wohlhart, Y. Bai, M. Kalakrishnan, S. Levine, and C. Finn, *Watch, try, learn: Meta-learning from demonstrations and rewards.*, *arXiv* (2019).

[20] N. Landolfi, G. Thomas, and T. Ma, *A model-based approach for sample efficient multi-task reinforcement learning.*, *arXiv* (2019).

[21] V. Mnih, K. Vavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, *Human-level control through deep reinforcement learning.*, *Nature* (2015).

[22] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning.*, *International Conference on Machine Learning* (2016).

[23] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, *Deterministic policy gradient algorithms.*, *International Conference on Machine Learning* (2014).

[24] R. Sutton, *Learning to predict by the methods of temporal differences.*, *Machine Learning* (1988).

[25] A. Nagabandi, C. Finn, and S. Levine, *Deep online learning via meta-learning: continual adaptation for model-based rl.*, *International Conference on Learning Representations* (2019).

[26] E. Friedman and F. Fontaine, *Generalizing across multi-objective reward functions in deep reinforcement learning.*, *arXiv* (2018).

[27] A. Abels, D. Roijers, T. Lenaerts, A. Nowe, and D. Steckelmaker, *Dynamic weights in multi-objective deep reinforcement learning.*, *International Conference on Machine Learning Research* (2018).

[28] D. Roijers and S. Whiteson, *Multi-objective decision making.*, *Synthesis Lectures on Artificial Intelligence and Machine Learning* (2017).

[29] B. Mitchell and L. Petzold, *Control of neural systems at multiple scales using model-free, deep reinforcement learning.*, *Scientific Reports* (2018).

[30] Y. Gao, E. Archer, L. Paninski, and J. Cunningham, *Linear dynamical neural population models through nonlinear embeddings.*, *Conference on Neural Information Processing Systems* (2016).

[31] M. Azar, R. Munos, and H. Kappen, *On the sample complexity of reinforcement learning with a generative model.*, *International Conference on Machine Learning* (2012).

[32] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, *Continuous deep q-learning with model-based acceleration.*, *International Conference on Machine Learning* (2016).

[33] D. Kappel, S. Bahenschuss, R. Legenstein, and W. Maass, *Synaptic sampling: a bayesian approach to neural network plasticity and rewiring.*, *Conference on Neural Information Processing Systems* (2015).

[34] Y. Loewenstein, A. Kuras, and S. Rumpel, *Multiplicative dynamics underlie the emergence of the log-normal distribution of spine sizes in the neocortex in vivo.*, *The Journal of Neuroscience* (2011).

[35] M. Pisauro, E. Fouragnan, C. Retzler, and M. Philiastides, *Neural correlates of evidence accumulation during value-based decisions revealed via simultaneous eeg-fmri.*, *Nature Communications* (2016).

[36] D. Kappel, R. Legenstein, S. Habenschuss, M. Hsieh, and W. Maass, *A dynamic connectome supports the emergence of stable computational function of neural circuits through reward-based learning.*, *arXiv* (2018).

[37] A. Lee, B. Nadler, and L. Wasserman, *Treelets-an adaptive multi-scale basis for unordered data.*, *The Annals of Applied Statistics* (2008).

[38] O. Gjata, M. Asllani, L. Barletti, and T. Carletti, *Using hamiltonian control to desynchronize kuramoto oscillators.*, *Physical Review E* (2016).

[39] M. Vittot, *Perturbation theory and control in classical or quantum mechanics by an inversion formula.*, *Journal of Physics A* (2004).

[40] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller, *Data-efficient deep reinforcement learning for dexterous manipulation.*, *arXiv* (2017).

[41] A. Ng, D. Harada, and S. Russell, *Policy invariance under reward transformations: theory and application to reward shaping.*, *International Conference on Machine Learning* (1999).

[42] B. Ziebart, A. Maass, J. Bagnell, and A. Dey, *Maximum entropy inverse reinforcement learning.*, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* (2008).

[43] D. Menell, A. Dragan, P. Abbeel, and S. Russell, *Cooperative inverse reinforcement learning.*, *Conference on Neural Information Processing Systems* (2016).

[44] M. Wulfmeier, P. Ondruska, and I. Posner, *Maximum entropy deep inverse reinforcement learning.*, *arXiv* (2016).

[45] C. Finn, S. Levine, and P. Abbeel, *Guided cost learning: deep inverse optimal control via policy optimization.*, *International Conference on Machine Learning* (2016).

[46] P. Koh and P. Liang, *Understanding black-box predictions using influence functions.*, *International Conference on Machine Learning* (2017).

[47] N. Heess, J. Hunt, T. Lillicrap, and D. Silver, *Memory-based control with recurrent neural networks.*, *Conference on Neural Information Processing Systems* (2015).

[48] D. Kingma and J. Ba, *Adam: a method for stochastic optimization.*, *arXiv* (2015).

[49] S. Haar, O. Donchin, and I. Dinstein, *Individual movement variability magnitudes are explained by cortical neural variability.*, *Journal of Neuroscience* (2017).

[50] H. Wu, Y. Miyamoto, L. Castro, B. Lvecsky, and M. Smith, *Temporal structure of motor variability is dynamically regulated and predicts motor learning ability.*, *Nature Neuroscience* (2014).

[51] B. Olveczky, A. Andalman, and M. Fee, *Vocal experimentation in the juvenile songbird requires a basal ganglia circuit.*, *PLOS Biology* (2005).

[52] M. Kao, A. Doupe, and M. Brainard, *Contributions of an avian basal ganglia-forebrain circuit to real-time modulation of song.*, *Nature* (2005).

[53] E. Tumer and M. Brainard, *Performance variability enables adaptive plasticity of 'crystallized' adult birdsong.*, *Nature* (2007).

[54] L. Kadanoff, *Statistical Physics: Statics, Dynamics and Renormalization.* 2000.

[55] G. Leal, *Advanced Transport Phenomena.* 2012.

[56] L. Landau and E. Lifshitz, *Quantum Mechanics.* 1965.

[57] H. Heekeren, S. Marrett, P. Bandettini, and L. Ungerleider, *A general mechanism for perceptual decision-making in the human brain.*, *Nature* (2004).

[58] B. Forstmann, G. Dutilh, S. Brown, J. Neumann, D. von Cramon, K. Ridderinkhof, and E. Wagenmakers, *Straitum and pre-sma facilitate decision-making under time pressure.*, *Proceedings of the National Academy of Sciences* (2008).

[59] L. Harrison, O. David, and K. Friston, *Stochastic models of neuronal dynamics.*, *Philosophical Transactions of the Royal Society* (2005).

[60] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, *Reinforcement learning with deep energy-based policies.*, *arXiv* (2017).

[61] M. Mattar, N. Wymbs, A. Bock, G. Aguirre, S. Grafton, and D. Bassett, *Predicting future learning from baseline network architecture.*, *Neuroimage* (2018).

[62] P. Ortega and D. Braun, *Thermodynamics as a theory of decision-making with information-processing costs.*, *Proceedings of the Royal Society* (2013).

[63] P. Ortega and D. Braun, *A minimum relative entropy principle for learning and acting.*, *Journal of Artificial Intelligence Research* (2010).

[64] H. Kappen, Y. Gomez, and M. Opper, *Optimal control as a graphical model inference problem.*, *Machine Learning* (2012).

[65] J. van den Broek, W. Wiegerinck, and H. Kappen, *Risk-sensitive path integral control.*, *Conference on Uncertainty and Artificial Intelligence* (2010).

[66] D. Braun, P. Ortega, E. Theodorou, and S. Schaal, *Path integral control and bounded rationality.*, *Adaptive Dynamic Programming And Reinforcement Learning* (2011).

[67] R. Adams, S. Shipp, and K. Friston, *Predictions not commands: active inference in the motor system.*, *Brain Structure and Function* (2013).

[68] K. Friston, S. Samothrakis, and R. Montague, *Active inference and agency: optimal control without cost functions.*, *Biological Cybernetics* (2012).

[69] K. Friston, J. Mattout, and J. Kilner, *Action understanding and active inference.*, *Biological Cybernetics* (2011).

[70] K. Friston, *The free-energy principle: a unified brain theory?*, *Nature Reviews Neuroscience* (2010).

[71] A. Moselhy and Y. Marzouk, *Reinforcement learning with deep energy-based policies.*, *Journal of Computational Physics* (2012).

[72] D. Bassett, N. Wymbs, M. Porter, P. Mucha, J. Carlson, and S. Grafton, *Dynamic reconfiguration of human brain networks during learning.*, *Proceedings of the National Academy of Sciences* (2011).

[73] D. Bassett, N. Wymbs, M. Rombach, M. Porter, P. Mucha, and S. Grafton, *Task-based core-periphery organization of human brain dynamics.*, *PLOS Computational Biology* (2013).

[74] A. Khambhati, M. Mattar, N. Wymbs, S. Grafton, and D. Bassett, *Beyond modularity: fine-scale mechanisms and rules for brain network reconfiguration.*, *NeuroImage* (2018).

[75] P. Reddy, M. Mattar, A. Murphy, N. Wymbs, S. Grafton, T. Satterthwaite, and D. Bassett, *Brain state flexibility accompanies motor-skill acquisition.*, *NeuroImage* (2018).

[76] A. Berry, V. Shah, S. Baker, J. Vogel, J. O'Neil, M. Janabi, H. Schwimmer, S. Marks, and W. Jagust, *Aging affects dopaminergic neural mechanisms of cognitive flexibility.*, *The Journal of Neuroscience* (2007).

[77] S. Fusi, W. Asaad, E. Miller, and X. Wang, *A neural circuit model of flexible sensori-motor mapping: learning and forgetting on multiple timescales.*, *Neuron* (2007).

[78] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Barwinska-Grabska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, *Overcoming catastrophic forgetting in neural networks.*, *Proceedings of the National Academy of Sciences* (2017).

[79] E. Neftci and B. Averbeck, *Reinforcement learning in artificial and biological systems.*, *Nature Machine Intelligence* (2019).

[80] J. Glascher, N. Daw, P. Dayan, and J. O'Doherty, *States versus rewards: Dissociable neural prediction error signals underlying model-based and model-free reinforcement learning.*, *Neuron* (2010).

[81] J. Glascher, R. Adolphs, H. Damasio, A. Bechara, D. Rudrauf, M. Calamia, L. Paul, and D. Tranel, *Lesion mapping of cognitive control and value-based decision making in the prefrontal cortex.*, *Proceedings of the National Academy of Sciences* (2012).

[82] R. Guo, W. Bohmer, M. Hebart, S. Chien, T. Sommer, K. Obermayer, and J. Glascher, *Interaction of instrumental and goal-directed learning modulates prediction error representations in the ventral striatum.*, *Journal of Neuroscience* (2016).

[83] W. Schultz, *Reward prediction error.*, *arXiv* (2018).

[84] C. Starkweather, B. Babayan, N. Uchida, and S. Gershman, *Dopamine reward prediction errors reflect hidden state inference across time.*, *Nature Neuroscience* (2017).

[85] B. Mitchell, N. Lauharatanahirun, J. Garcia, N. Wymbs, S. Grafton, J. Vettel, and L. Petzold, *A minimum free energy model of motor learning.*, *Neural Computation* (2019).

[86] V. Pong, S. Gu, M. Dalal, and S. Levine, *Temporal difference models: Model-free deep rl for model-based control.*, *International Conference on Learning Representations* (2018).

[87] M. Marneweck, D. Barany, M. Santello, and S. Grafton, *Neural representations of sensorimotor memory- and digit position-based load force adjustments before the onset of dexterous object manipulation.*, *Journal of Neuroscience* (2019).

[88] J. Diedrichsen and R. Shadmehr, *Detecting and adjusting for artifacts in fmri time series data.*, *Neuroimage* (2005).

[89] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, *Hindsight experience replay.*, *International Conference on Neural Information Processing Systems* (2017).

[90] M. Bellemare, W. Dabney, and R. Munos, *A distributional perspective on reinforcement learning.*, *arXiv* (2017).

[91] M. Lindquist, J. Loh, L. Atlas, and T. Wager, *Modeling the hemodynamic response function in fmri: efficiency, bias, and mis-modeling.*, *Neuroimage* (2008).

[92] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya, and T. Tanaka, *Nonparametric return distribution approximation for reinforcement learning.*, *International Conference on Machine Learning* (2010).

[93] Z. Kurth-Nelson and A. Redish, *Temporal-difference reinforcement learning with distributed representations.*, *PLOS One* (2009).

[94] J. Zhang, L. Hughes, and J. Rowe, *Selection and inhibition mechanisms for human voluntary action decisions.*, *NeuroImage* (2012).

[95] B. Kim and M. Basso, *A probabilistic strategy for understanding action selection.*, *Journal of Neuroscience* (2010).

[96] J. Gallivan, L. Logan, D. Wolpert, and J. Flanagan, *Parallel specification of competing sensorimotor control policies for alternative action options.*, *Nature Neuroscience* (2016).

[97] S. Schernaum, M. Dshemushadse, R. Rischer, and T. Goschke, *How decisions evolve: the temporal dynamics of action selection.*, *Cognition* (2010).

[98] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, and M. Riedmiller, *Data-efficient deep reinforcement learning for dexterous manipulation.*, *arXiv* (2017).

[99] A. Ng, D. Harada, and S. Russell, *Policy invariance under reward transformations: theory and application to reward shaping.*, *International Conference on Machine Learning* (1999).

[100] T. Kulkarni, K. Narasumhan, A. Saeedi, and J. Tenenbaum, *Hierarchical deep reinforcement learning: integrating temporal adbstraction and intrinsic motivation.*, *Conference on Neural Information Processing Systems* (2016).

[101] S. Krishnan, A. Garg, R. Liaw, L. Miller, F. Pokorny, and K. Goldberg, *Hirl: Hierarchical inverse reinforcement learning for long-horizon tasks with delayed rewards.*, *arXiv* (2016).

[102] D. Hernandez-Lobato, J. Hernandez-Lobato, A. Shah, and R. Adams, *Predictive entropy search for multi-objective bayesian optimization.*, *International Conference on Machine Learning* (2016).

[103] E. Garrido-Merchin and D. Hernandez-Lobato, *Predictive entropy search for multi-objective bayesian optimization with constraints.*, *arXiv* (2016).

[104] A. Shah and Z. Ghahramani, *Pareto frontier learning with expensive correlated objectives.*, *International Conference on Machine Learning* (2016).

[105] S. Suzuki, S. Takeno, T. Tamura, K. Shitara, and M. Karasuyama, *Multi-objective bayesian optimization using pareto-frontier entropy.*, *arXiv* (2019).

[106] K. Moffaert and A. Nowe, *Generalizing across multi-objective reward functions in deep reinforcement learning.*, *Journal of Machine Learning Research* (2014).

[107] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley, *Generalizing across multi-objective reward functions in deep reinforcement learning.*, *Journal of Artificial Intelligence Research* (2013).

[108] B. Gebken and S. Peitz, *Inverse multiobjective optimization: inferring decision criteria from data.*, *arXiv* (2019).

[109] S. Peitz and M. Dellnitz, *Gradient-based multiobjective optimization with uncertainties.*, *arXiv* (2017).

[110] A. Fermin, T. Yoshida, M. Ito, J. Yoshimoto, and K. Doy, *Evidence for model-based action planning in a sequential finger movement task.*, *Journal of Motor Behavior* (2010).

[111] A. Fermin, T. Yoshida, J. Yoshimoto, M. Ito, S. Tanaka, and K. Doy, *Model-based action planning involves cortico-cerebellar and basal ganglia networks.*, *Scientific Reports* (2016).

[112] M. Fazel, R. Ge, S. Kakade, and M. Mesbahe, *Global convergence of policy gradientmethods for the linear quadratic regulator.*, *arXiv* (2019).

[113] J. Taylor and R. Ivry, *The role of strategies in motor learning.*, *Annals of the New York Academy of Sciences* (2012).

[114] N. Wymbs and S. Grafton, *The human motor system supports sequence-specific representations over multiple training-dependent timescales.*, *Cerebral Cortex* (2015).

[115] R. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, *Neural ordinary differential equations.*, *International Conference on Neural Information Processing Systems* (2018).

[116] J. Wang, Z. Kurth-Nelson, D. Kumaran, D. Tirumala, H. Soyer, J. Leibo, D. Hassabis, and M. Botvinik, *Prefrontal cortex as a meta-reinforcement learning system.*, *Nature Neuroscience* (2018).

[117] S. Grafton, E. Hazeltine, and R. Ivry, *Functional mapping of sequence learning in normal humans.*, *Journal of Cognitive Neuroscience* (1995).

[118] E. Hazeltine, S. Grafton, and R. Ivry, *Attention and stimulus characteristics determine the locus of motor-sequence encoding: A pet study.*, *Brain : A Journal of Neurology* (1997).

[119] S. Grafton, E. Hazeltine, and R. Ivry, *Abstract and effector-specific representations of motor sequences identified with pet.*, *The Journal of Neuroscience* (1998).

[120] G. Coricelli and R. Nagel, *Neural correlates of depth of strategic reasoning in medial prefrontal cortex.*, *Proceedings of the National Academy of Sciences* (2009).

[121] J. Decety, P. Jackson, J. Sommerville, T. Chaminade, and A. Meltzoff, *The neural bases of cooperation and competition: an fmri investigation.*, *NeuroImage* (2004).

[122] D. Lee, *Game theory and neural basis of social decision making.*, *Nature Neuroscience* (2008).

[123] K. Ahmed and A. Jelodar, *Fine-tuning vgg neural network for fine-grained state recognition of food images.*, *arXiv* (2018).

[124] J. Howard and S. Ruder, *Universal language model fine-tuning for text classification.*, *arXiv* (2018).

[125] A. Radford, R. Wu, J. Child, D. Luan, D. Amodei, and I. Sutskever, *Language models are unsupervised multitask learners.*, *International Conference on Machine Learning* (2018).