Title
Inference Algorithms and Sensorimotor Representations in Brains and Machines

Author
Mudigonda, Mayur

Publication Date
2019

Peer reviewed|Thesis/dissertation

Inference Algorithms and Sensorimotor Representations in Brains and Machines

by

Mayur Mudigonda

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Vision Science

and the Designated Emphasis

in

Communication, Computation and Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Bruno A Olshausen, Chair
Professor Michael R Deweese
Professor Jitendra Malik
Professor Pieter Abbeel

Summer 2019

Inference Algorithms and Sensorimotor Representations in Brains and Machines

Abstract

Inference Algorithms and Sensorimotor Representations in Brains and Machines

by

Mayur Mudigonda

Doctor of Philosophy in Vision Science
and the Designated Emphasis in
Communication, Computation and Statistics

University of California, Berkeley

Professor Bruno A Olshausen, Chair

Animals function in a 3D world in which survival depends on robust, well-controlled actions. Historically, researchers in Artificial Intelligence (AI) and neuroscience have explored sensory and motor systems independently. There is a growing body of literature in AI and neuroscience to suggest that they actually work in tandem. While there has been a great deal of work on vision and audition as sensory modalities in these fields, one could argue that a more fundamental modality in biology is haptics, or the sense of touch. In this thesis, we will look at building computational models that integrate tactile sensing with other sensory modalities to perform manipulation-like tasks in robots and discrimination tasks in mice. We will also explore the problem of inference through the lens of Markov Chain Monte Carlo methods (MCMC). We elaborate on the ideas discussed in this thesis in the introduction presented in Chapter 1.

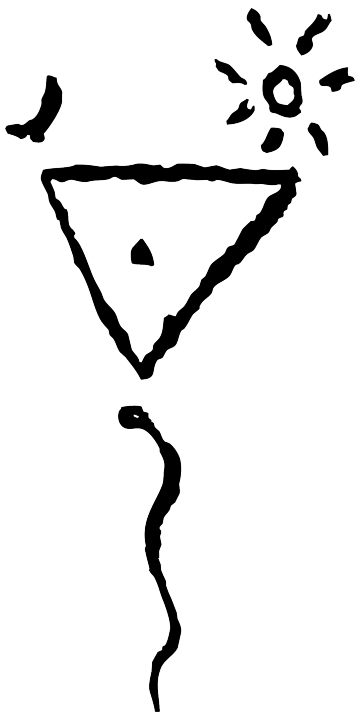
A challenging problem one often faces when applying probabilistic mathematical models to the study of sensory-motor systems and other problems involving learning of inference is sampling. Hamiltonian Markov Chain Monte Carlo (HMC) algorithms can efficiently draw representative samples from complex probabilistic models. Most MCMC methods rely on detailed balance to ensure that we can sample from the correct distribution. This constraint can be relaxed in discrete state spaces such as those employed by HMC type methods. In Chapter 2, we study HMC methods without detailed balance to explore faster convergence. Markov jump processes are stochastic processes on discrete state space but continuous in time. In Chapter 3, we use Markov Jump Processes to simulate waiting times along with generalized detailed balance. This waiting time ,we show, helps generate samples faster. Most MCMC methods are plagued by slow simulation times on discrete computing systems. In Chapter 4, we explore HMC in analog circuits where the problem of generating samples from a distribution is mapped to the problem of sampling charge in a capacitor.

The second half of this dissertation focuses on the role of haptics in perception and action. Manipulation is a fundamental problem for artificial and biological agents. High

dimensional actuators (say, fingers, trunks,etc) are really hard to control. In Chapter 5, we present an approach to learn to actuate dexterous manipulators to grasp objects in simulation. Haptics as a sensory modality is critical to many manipulation tasks. Employing haptics in high dimensional dextrous actuators is challenging. In Chapter 6, we explore how intrinsic curiosity and haptics can be used to learn exploration strategies for discrimination of objects with dextrous hands. A key component to make tactile sensing a possibility is the availability of cheap, efficient, scalable hardware. Chapter 7 presents results for tactile servoing using a physical gelsight sensor. Traditional neuroscience texts delineate sensory and motor systems as two independent systems yet recent results suggest that this may not be entirely complete. That is, there is evidence to suggest that the representations in the cortex is more distributed than is accepted. Finally in Chapter 8, we explore building a computational model of spiking neural data collected from both the barrel and motor cortices during free and active whisking. These works help towards understanding sensorimotor representations in the context of haptics and high dimensional controls. We conclude with a discussion on future directions in Chapter 9.

I dedicate this work to my guru, *Sadhguru*.

Shambo



Contents

List of Figures

List of Tables

Acknowledgments

As the drop is part of the ocean, so is the ocean a part of the drop

I wish to acknowledge the very many people who made this journey possible. I hope I can, to the best of my abilities, include all those who have contributed towards this endeavor and I apologize to any whom I have missed.

First, my family. I would like to thank my mother for nourishing me and bringing me into this world. For teaching me discipline, creativity and teaching me to take care of myself. I would also like to thank My father for instilling in me perseverance, following my dreams, and the power of understanding through simplicity. Last but not least, I would like to thank my brother for constantly pushing me beyond my limitations and for timely advice. I am also grateful to my extended family for their many roles in my life. Words cannot express my thanks to them. They helped me shape who I am.

I have had the absolute privilege of being in the company of many whom I consider friends. Calling them friends severely downplays their role in my life as they have been nothing short of phenomenal inspiration, soothing comfort and a source of wisdom.

I would like to thank my friend Aditya Ramani (Adi) for being a fantastic support in my life spanning now over a decade and a half. For always providing me honest feedback in a loving, humorous way. Most importantly for the care and love that I receive(d).

My thanks to Karthik Ganesh for exposing me to so many philosophies and ideas and for the countless conversations that have helped me reflect and shape who I am.

My love and thanks to Pavan Ramkumar, with whom I have shared a long friendship that dates back to my days in kindergarten. We have shared an adventure of science, exploration, debate that weaves in and out of neuroscience, computer science, philosophy and the arts. These have had a profound impact on my life. I am grateful for all of these.

My love and gratitude to Arjun Krishnan whose brilliance was always tempered with a deep love and care . It has been a privilege learning to grow with and from you. My thanks to Cheenu for showing gracefulness in all situations and for always having a welcoming place for me to crash when I needed it the most.

I am very grateful to my mentors in India - Profs. Sengathir and Vinay thank you for showing me how to be absorbed in a problem while having a humorous disposition. They were the first teachers who helped me start my journey of research.

A special thanks to my friends' moms (and dads) who fed me during my undergraduate studies, I thank you!. My love to Thangam Paati for adopting me as her grandson and looking after me with her infectious warmth. My gratitude to Vick(ed)ram and Chitra aunty for all the wisdom, wit and love. My love to Sunita Auntie for feeding me delicious meals and sharing how to care. Special thanks to Meera auntie for the very many delicious meals that I enjoyed.

To my friends during my time at Michigan State University, I thank you. I am very grateful to my friend Naveen for teaching me to believe in myself, for always having a comforting shoulder to lean on when I needed it the most, nurturing my scientific progress

and also for helping me switch to Linux! My undying friendship to Zubin Abraham for showing me how to find the best in every human being in every situation and for the endless late night conversations on all things science, art and more. My love to Farhana Khan for showing me how to balance responsibility with humor and love. My gratitude to Pavan Mallapragada for teaching me many things about machine learning, computer vision and guitars. I would like to thank Kantha Pongaliur and Ashsish Dore for being supportive in my formative years as a scientist. To my friend Karthik Hemmanur, thank you for your support, laughs and adventures over the years.

To my mentors at Michigan State University - Prof. Stockman, Jain and Jin, my thanks. I would like to thank Prof.Stockman for constantly encouraging me to understand the basics of 3D vision and for teaching me how to communicate more effectively. My thanks to Prof.Rong Jin for teaching me the basics and the joys of machine learning. I am grateful to Prof.Anil Jain for teaching me the value of being a rigorous scientist, communicating effectively and being a great mentor. My time at MSU was also shaped by my teaching experience with Jo (Smith). I would like to thank Jo for teaching me the importance of a strong work ethic and encouraging me to pursue my dreams. To the various members of the CogSci group you have my gratitude.

To my friend Vidhya, I value our friendship and time together. It has helped shape how I think. To my friend Deepika, thank you for inspiring me and for being a good listener.

I would like to thank Jack Gallant for inviting me to Berkeley along with Sonia to explore experimental neuroscience. It has been a great ride ever since. My time at Blindsight was a wonderful experience where I learnt about building systems that help people. I thank Mark Nitzberg, Peter Hallinan and Alan Yuille for their mentoring. I also thank Kaolin and Kevin for helping me learn how to code and grow.

I benefited immensely from the friendship and mentorship of David Dmerdjian, Wes Chaney, Summer Shermata, Jason Fischer, Britta Humel and Gerrit Maus during this time. They were the bedrock during my earliest days at Berkeley exposing me to things both professionally and personally (such as snow boarding!)

My time at Berkeley has been a truly remarkable, fun, adventerous and unique experience. I am deeply indebted to Bruno Olshausen, without whom I would never have done my PhD at Berkeley. His patience, kindness, inexhaustible source of knowledge, willingness to learn, to understand without judgement are unparalleled. I thank him for being a really good friend, mentor, and hope to contribute back to the community in a similar vein. I would also like to thank Mike whose optimism, cheerfulness and wisdom have been a source of great comfort and growth during my time here. It has been a joy to learn to work on problems with abandon with such mentors.

I would like to thank Fritz Sommer for helping me to learn to think deep about problems in computational neuroscience . My love to Tony Bell for the many many conversations on all things life, science and meditation. For showing me how to be a scientist in all situations and to be in touch with one's emotions. I am also honored to have spent time learning from Pentti Kanerva. Pentti has been a true inspiration to generations of scientists. I always enjoyed chatting and learning from Pentti on all things brains, machines and computation.

Last but not least I have enjoyed and benefitted from the company of Kris Bouchard during the later part of my PhD.

I am grateful to my friend Allie Fletcher who helped me understand the naunces of science, engineering and computational neuroscience. Allie was also a phenomenal support during the earliest part of my PhD. I can honestly say I would not have made it without her support. I would like to thank Asako for the countless conversations on all things science and art. I would also like to thank Natalia, who was the first friend I made when I moved to Berkeley who continues to inspire me with her quick, insightful observations whenever we hang out.

I would like to thank my friend Nisha Giridhar for being such an incredible support during the darkest days of my time here.

To my unofficial advisors my first year at the Redwood center Jascha Sohl-Dickstein, Urs Koster, Ian Stevenson, and Chris Hillar thank you so much! To Jascha for showing me how to think like a scientist, to be rigorous , to code and to think about math and physics. In addition to helping me learn to combine science with art and more, thanks Jascha! To Urs for having so many incredible adventures that were inside and outside the classroom discussing all things science , computation and more - thank you! To Ian, for showing the importance of picking the right problem and showing me how to balance personal life and work - thank you. To Chris Hillar for showing me the beauty of Math, theoretical neuroscience and friendship - thanks!

I would like to acknowledge my thesis committee here as well who have played a significant role in educating me and shaping my world view of perception and action. To Jitendra for always taking the time out to meet with me and for being a walking, talking encyclopedia of information. Most importantly, reminding me that *those who do not know their history are doomed to repeat it* - thank you Jitendra! To Pieter, whose robotics class was very significant in my understanding of controls and for also wanting to make a foray into sensorimotor representations. I would also like to thank Pieter to help me learn to manage doing science that straddles multiple disciplines. I would also like to thank Alyosha who has been an unofficial mentor to me and many alike in the BAIR community. Your keen perception, rigorous questioning always interspersed with nuggets of wisdom and humor greatly enriched working late nights with other BAIR folks. I greatly enjoyed our sensorimotor rendezvous with chocolates and conversations into the wee hours of the morning.

I have benefitted immensely from many conversations with Sergey on robotics, neuroscience and machine learning. My conversations with Roberto greatly shaped my thinking about problems in robot learning and working with hardware. I greatly enjoyed and benefited from the many conversations with Dinesh on vision, haptics, meditation and life. To my collaborators Frederik, Stephen, Chelsea thank you for showing me how to work on various problems in robotics.

To my friends and cohort at the Vision Science community, thank you so much for making me a part of you and your community. The wisdom and love of this community is simply outstanding. I would also like to acknowledge my faculty mentor Michael Silver for being such an inspiration during my time at Berkeley. I always came away from those

meetings learning more about science and myself. I am grateful for his time, effort, patience and kindness. My thanks to Angie, Carissa and Natalie for being helpful in navigating the various administrative processes at Berkeley.

To Chris Warner, Zayd Enam , Yubei Chen, Eric Weiss, Vasha Dutell, Spencer Kent, Dylan Paiton, Shariq Mobin, Brian Cheung, Guy Isley, Ryan Zarcone, Mike Fang, Charles Frye, Mike Schachter, and the countless other redwood visitors, scientists and friends thank you from the bottom of my heart. The redwood center would not have been colorful without you and my time would not have been as deeply enriched. Thank you for being who you are and for helping me engage so deeply in the various aspects and studies of our fields and life.

To my office mates when I moved into a smaller office Jesse Livezey and Joe Thurakal, my gratitude. We forged such a brilliant friendship and had so many adventures that I will always fondly remember. It has been an absolute privilege to learn humility, set internal metrics for growth and contribution and find my feet again when I thought the world had gone dark. I am infinitely grateful for yours and your significant others' (Jenn, Sarah) presence in my life.

To my more recent office mates Louis and Neha, thank you so much for the various conversations over the last few years. The afternoon radio stations leaning towards classical music, sharing with me a better way to balance work priorities along with a burning curiosity to understand things - thank you for the wonderful times!

Spending a summer at Flickr research was a great way for me to learn and grow. I am deeply grateful for the opportunity and would like to acknowledge my mentors Jack Culpepper and Simon Osindero. They spent a lot of time helping me learn so many things and also inspiring me to understand the nuances of machine learning, deep learning and computer vision. The rest of the Flickrenos were also incredibly supportive and helpful during my time there.

The last few years I have been deeply involved in exploring the connections between meditation and neuroscience with the Isha Research Center. This has been a phenomenal experience of growth and learning. It has also been some of the most exciting research experiences I have had in my career so far. I would like to thank Suresh, Bala, Senthil, Raj, Ramana and Sepideh for this opportunity and look forward to incredible adventures in the future.

To the Woodhouse crew thank you for making my time here so enjoyable and amazing. To Aadi for being the center of all our activities (Friday dinners), adventure (to LA, Africa and more!) and being such an inspiration to me. You have singularly shaped my life in such a significant way. My love and thanks to Leila for being so perceptive, skeptical and willing to engage intellectually. To Siva, Vish, Nishant, Madhyama, and other guests - thank you for the company, food and warmth.

I am also indebted to my friend Avinash for teaching me the merits of perseverance and for letting me vicariously enjoy being an ultra ultra marathon crew person. To my friend Christian Weber for including me in his life and having many conversations about physics, philosophy and life.

To Dobby and Winky, for the endless support, food, conversations,laughs and love. I thank you from the bottom of my heart along with my gratitude! You are now free elves. *Please to be free.*

To my longest and dearest collaborator Pulkit. For the many late night conversations on AI, neuroscience and life, I thank you. Those conversations greatly shaped how I think about science and will continue to shape how I think about life and science. To the rest of the BAIR community – Evan Shelhammer, Saurabh Gupta, Shiry Ginosar, Richard Zhang, Deepak Pathak, Chelsea Finn, Rachit Dubey, Shubham Tulsiani. Thank you!

To the very many undergrads with whom I have had the privilege of learning through teaching and growing, I thank you from the bottom of my heart. Aditya for being a wonderful student and teaching me so many things about Jazz. Shayan for teaching me about hardwork and willingness. To Blake with whom I learnt how to balance learning, hardwork, and play. To Andrew for giving me an opportunity to learn more things about statistics and physics than I knew before. To Ankur for being such an incredible and enthusiastic collaborator and educating me on various things related to climate science. To Jiayi and Kevin for helping build the climateNet tool. I hope I have done the best I can as a mentor and I look forward to your future successes.

My deepest thanks and gratitude to Prabhat for really providing a platform for my research and including me in various projects at the Berkeley National Lab. I am also grateful to Prabhat for teaching me how to be an effective manager and for the many fun games of badminton. I learnt a great deal about high performance computing and tensorflow from Thorsten Kurth, Mustafa and the rest of the Gordon Bell prize winning team, my congratulations and many thanks for the opportunities to learn. I also would llke to thank my friends and mentors Karthik Kashinath, Adrian Albert for the very many multi-lingual science and life conversations. My thanks to the other members of the DAS team! And my gratitude to Paolo and Steve for the adventures in high energy physics and deep learning.

My time at Berkeley was also shaped by my friends at Cal badminton - Tuomas Koskela, Sayan Gupta, Deepak Talwar, Jane Pan, Manu Sharma, Ali, Fumika and others greatly helped make badminton and Cal a lot of fun!

During my time at Berkeley I have had the privilege of doing the Aids Life Cycle ride from SF to LA. This was an incredible experience and I would like to thank the very many who made this possible. Specially, during those long training rides it was a comfort to have Marie cheering us on and sharing with us food and drinks. I also had the privilege of biking through Zambia as part of Bike Zambia and I would like to thank Clair and her crew for really including us and being so warm, willing and endearing. To the rest of the riders, it was a truly memorable experience.

I cannot ever express the role meditation has played in my life. Most importantly, the role *Sadhguru* has played in my life. Thank you from the bottom of my heart *Sadhguru* in helping me break my limitations and blossoming into a joyful human being. I do not have sufficient words to express my gratitude. To the many Ishas who are but kin to me now, and always - my gratitude. To the Ishaangas who have helped me in so many ways - Gomati, Yemuna, Maya, Ganesh, Senthil and Aparna. Thank you. To the residents of III - Shanti,

Saloney, Ramez, Rajat, Manisha, Leah, Swamis Ibana, Harsha, Chitranga and the countless other Ishas, thank you so much for showing me how to be.

My love and thanks to my friend Partha for being a willing, loving ear to the very many teething troubles I have had (and continue to have) while learning to meditate and navigate graduate school. To Sepideh for her love, inspiration and being the bedrock in my life. My love and thanks to Nishi, Sandeep and Sempa for being so giving, sharing adventures and in taking care of me. My gratitude to Ankur Jain for showing me that it is possible to be a volunteer in every situation. My deepest love to Sangeeta akka who showed me how to give without asking anything in return. Special thanks to Shailendra for inviting me to so many of his house parties and plying me with amazing food. My gratitude to Surya and Ramana who showed me how to be involved and manage my time better without compromising on being loving. To Sangeeth, who helped me understand how to caring about oneself through the lens of meditativeness and humor. My deepest love and gratitude to Rami who completely transformed how I think and helped me accept things the way they are. My thanks to Srividhya for bringing about an infusion of love and joy into my life.

My gratitude to Ravishankar Nilekantan and Ram Krishnan for showing me what it means to volunteer with intensity, involvement while also managing a challenging work commitment.

Last but not least my undying love and gratitude to Mai and the Red Dot(ties). Mai thank you for being you and for being a mother to me in more ways than I can imagine.

Nischala Tatvam Jeevan Muktihi

Chapter 1

Introduction

Our brain is an organ that is directed towards action

- Ramon y Cajal

It has been argued by philosophers and scientists that the goal of the brain is to enable action in the world. This argument should then naturally lend itself to the fact that sensory representations should be geared towards acting in the world [62, 145].

Biology solves the problem of action and perception in an overactuated motor system and a multimodal-sensory system with a tight coupling between the two systems. What is particularly fascinating with the way biology solves this problem is that the agent has an understanding of self and the world. Compelling examples toward this comes from the following:

a) An agent is able to discriminate its percept to delineate self and world motion b) An agent has understanding of its body parts with respect to its sensors c) An agent has an understanding of how its actions will effect itself and the world

Can we build computational models for learning sensorimotor representations which can be applied to both robotics and to explain biological circuitry? These are the problems that motivate my research.

In their seminal work Hubel and Wiesel [87, 86] showed that sensory neurons are orientation selective (with varying degrees of spatial invariance). This inspired a computational framework to extract such high frequency orientations (center-surround edges) in an invariant way as proposed by Marr and Hildreth [129].

While many extensions to this model were proposed by scientists (for e.g. [26]) to extract edges, these tended to ignore the statistics of the data (more specifically natural scenes). Another line of thinking was proposed based on efficient coding and redundancy reduction principles as proposed by Barlow [10].

Early work in this was motivated by Field [51], who showed that the power spectrum of natural images had a specific structure. Attick and Redlich [7] proposed a mechanism to whiten the power spectrum of the data. They proposed an idea of center-surround filters to do this with evidence that perhaps the retina solved a similar problem. Inspired by

Barlow, Olshausen and Field proposed a model to find linear filters that are sparsely active [143]. The filters extracted by their model better matched the experimental data seen from physiologists studying early visual cortex such as V1. Similar filters could also be extracted by maximizing independence of filters as well, as was proposed by Bell and Sejnowski[12].

Yet residual statistical dependencies were found in these models. To model those dependencies hierarchical approaches were taken. For example, Cadieu and Olshausen [20] tried to model both form and motion in a factorial generative model with complex basis functions.

While research continues to progress along those lines of thinking, it still leaves something to be desired, in that, these image coding models are not primarily geared towards enabling an agent to act in the world. For example, it is important to note that many of these models are trained with camera images and thus the representation is at best camera centric. Thus, many more complicated tasks such as grasping, locomotion, eye movements that require a tight coupling with the sensory representations may not benefit from the image coding models as-is without a more explicit understanding and modeling of the visuo-motor transformations.

Research on the human visual system suggests that we have separate representations of the 3D world that are often described as the what and where streams. Perhaps one way to think of computational models we have looked at above could be thought to be computational models of the what stream. So what maybe the contributions of the where stream for an agent?

What is lacking, from a computational perspective is a theoryIn that elegantly links sensory and motor systems. The foundations of such a theory has been proposed by O'Regan and colleagues [12,13] who showed that in the case of compensable motion the dimensionality of the sensorimotor manifold is smaller than the dimensionality of the sensory or motor manifold individually.

What evidence might there be?

Some remarkable work by Melvyn Goodale and colleagues [69, 70] have shown that patient DF [1] while unable to identify or discriminate objects could still shape and orient her grasp accurately for an object. This seems to suggest a parallel distributed coding of the visual world: one with a complex semantic representation and another geared more toward action with a more metric representation of space.

Goodale et al. [69] proposed the two stream hypothesis suggesting that there exists two streams of processing in the cortex. These can be viewed as the dorsal and the ventral stream. The ventral stream is though to process object identity, semantics and other perceptual features. The dorsal stream in contrast is thought to maintain a more euclidean-like representation of the world. Indeed, the studies shown with patient DM [70] show that grasp sizes are not affected by perceptual confounds even though stimuli used would confound the percept (as seen in Figure 1.1)

[1]Patient DF is diagnosed with lesions in her ventral stream. The resulting inability to semantically identify an object is diagnosed as visual agnosia

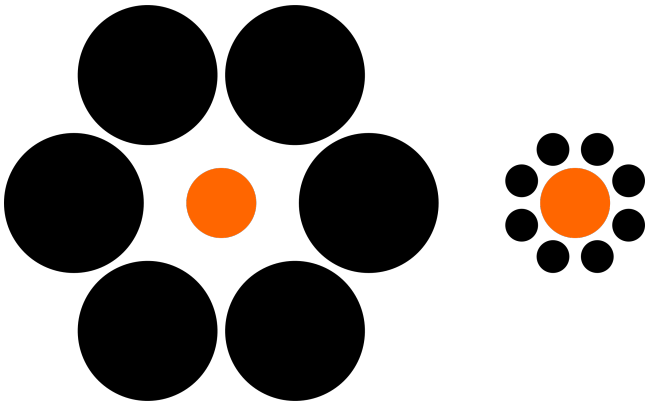


Figure 1.1: *Tichener circles illusion* in which two target circles of equal size, each surrounded by a circular array of either smaller or larger circles, are presented side by side. Subjects typically report that the target circle surrounded by the array of smaller circles appears to be larger than the target surrounded by larger circles. In test, two thin 'poker-chip' discs were used as the target circles. The relative size of the two discs was randomly varied so that on some trials the discs appeared perceptually different but were physically equivalent in size, and on other trials they were physically different but appeared perceptually equivalent. The perceptual judgements made by the 14 subjects in our experiment were strongly affected by this size-contrast illusion. However, when asked to pick up a disc, the scaling of the subjects grip aperture (measured opto-electronically before contact with the disc) was largely determined by the true size of the target disc and not its illusory size. Results from Aglioti et al. [2]

Most computational models can be thought to model the ventral stream. However the information in the dorsal stream is required for an agent to survive. In this thesis, we make an attempt towards such modeling.

There are many other compelling examples from experimental neuroscience that begs for a more comprehensive theory of representations. Guillery and Sherman in their works [171] have argued for a more recurrent role of the connections between the cortex and the thalamus - suggesting a very tightly coupled sensorimotor representation even as early as V1, V2.

Another example comes from the work of Niell and Stryker [141] who showed that the firing rate of V1 neurons in the mouse cortex more than doubled for the same stimulus when the mouse was in motion, thus arguing for a more coupled representation. See Figure 1.2 .

Recent advances in multielectrode recordings from multiple regions have led to some exciting discoveries. For example, [131] show that while activity in the motor cortex results in exploration of whiskers, stimulation in the sensory cortex results in a fast retraction of whiskers thus suggesting a need to reevaluate the functional organization of cortical maps.

In Chapter 8, we explore this particular problem of understanding the computational underpinnings in the barrel and motor cortices. We begin with first trying to fit a popula-

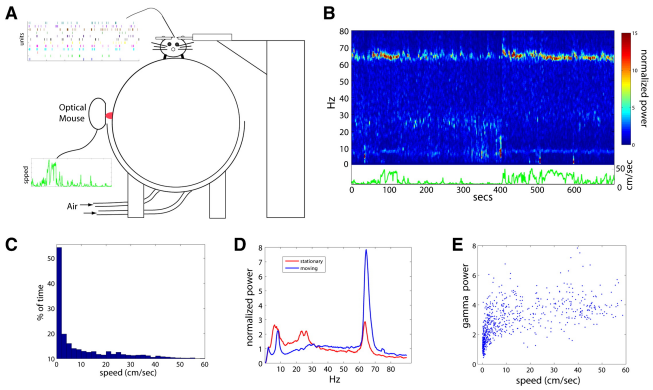


Figure 1.2: (A) The mouse's head is fixed on top of a styrofoam ball suspended by air. Multisite silicon probes are used to measure spiking units, while data from pairs of optical mice are used to calculate the motion of the ball under the mouse. (B) Local field potential (LFP) power during the duration of a single recording, with corresponding speed trace shown below in green. (C) Distribution of mouse speed, showing a large fraction of time spent stationary and a wide distribution of running speeds. (D) Average power spectrum from recording shown in (B), during stationary versus moving periods. (E) Scatter plot of power around gamma peak (60–70 Hz) versus speed of movement, demonstrating a sharp transition between stationary and moving states. From Niell and Stryker [141]

tion model using *Ising*models. We show that the learnt couplings actually show increased inhibitory activity during active whisking compared to free whisking (in the barrel cortex vS1) and in the motor cortex (vM1) we only see a reorganization.

We then look to understand the role temporal signaling plays in this circuit and fit models that capture the time varying statistics better. We show that we are able to predict physical features such as whisker angle, phase , velocity, etc particularly well with a recurrent network model.

We have motivated existence of a distributed representation that seems driven by action but what might be the ecological action space that is most natural to study?

Gibson for example [62, 61, 60] stated that representations are geared towards actions and proposed a theory of vision that dealt with *affordances*. Others have looked into navigation as a problem to consider as it lends itself to unsupervised learning.

Learning to manipulate

In our research, in chapter 5, we explored the problem of learning to actuate dextrous manipulators. We picked this problem primarily because it lends itself to self-supervision and provides an avenue to study multi-sensory integration.

Classical control requires a prespecified or a learnt dynamics model for actuation. This becomes prohibitive to do with high dimensional spaces (both with observations and actuators) with traditional methods. In recent times we have seen progress in high dimensional

reinforcement learning (RL) methods both in continuous spaces and high dimensions [135, 168]. In our work, we expand on this work by showing that we can teach a robotic, high dimension, dextrous manipulator to" grasp objects in simulation.

What role does haptics or tactile sensing play?

While different animals are equipped with different sensory modalities the sense of touch is in fact far more ubiquitous than any other modality. There is anecdotal evidence to suggest that tactile sensing plays a very critical role in manipulation. [96] show that when tactile sensing is inhibited using a local anesthetic the subject struggls to light even a simple matchstick.

In chapter 6 we explore how tactile sensing could be used to learn to discriminate objects. Exploration is an important part of reinforcement learning methods because they determine how an agent must explore its environment to maxmimize an objective.

There are many types of exploration strategies that border from heuristics to maximizing entropy. In our work we explored the idea of intrinsic curiosity [150] where we use the latent embedded prediction error as reward. That is, we compare the difference between predicted future state and the actual future state using a learnt latent embedding. This reward can then be combined with an extrinsic reward that is more task specific.

Role of sensors

Most simulated tactile sensors compare unfavorably to real tactile sensing. For example, [115] show the various hand movements we use to explore using tactile sensing. If we are ever to equip robots with the ability to do more sophisticated actions it would be important to equip them with tactile sensors that can approximate human sensing to some extent.

In chapter 7, we explore the use of the gelsight sensor [199] for servoing using model predictive control [53]. GelSight is an optical-based tactile sensor that uses a piece of coated elastomer as the contact medium. A camera records the distortion of the elastomer during contact.

We attach the gelsight sensor to a CNC machine, we then collect samples where an object is in contact with the sensor and is manipulated. We then train a model that can predict future visual (tactile) states given a sequence of initial frames. We use this mode to plan a policy to navigate a given object to a desired goal position.

Solving Perceptual Inference Problems with MCMC Sampling

The problem of perception is often thought of as the problem of inference [120]. For example, in the Figure 1.3 shown below- how do we go from seeing an image (a collection of pixels) to answering complicated questions such as identifying the person, their pose, the position of the sun and other semantic questions?

Figure 1.3: Factoring shape, shading and reflectance. Image courtesy of Ashwin Mudigonda ©

If we just searched the space of pixel combinations to explain the position of light in the image, the sheer combinatorics of the problem makes it untenable [2] to find a solution. One elegant way to deal with this problem is by using Bayesian statistics which encodes prior knowledge of how light works, to make inference easier [102, 103]

Bayes theorem states that the likelihood is combined with the prior to give us the posterior distribution. In the equation below $P(\theta)$ denotes the prior, where θ denotes parameters of the model and $P(X|\theta)$ the likelihood. Often times, we are interested in knowing what are the likely parameters that could have generated the sample X.

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)} \tag{1.1}$$

Now, if we were trying to estimate the parameter θ based on a sample X using a greedy optimization algorithm we may end up settling in a mode that is not entirely optimal. For example in Figure 1.4 one can see the mode on the left being less optimal than on the right. [98] provides a way to approximate complex posteriors using a variational approximation. While this is quite a popular way to express the distribution it does not fully represent the distribution and limits the expressive power of probabilistic models.

Thus, proper Inference in a complex Bayesian model can only be done using sampling or known as Markov Chain Monte Carlo methods (MCMC). The Markovian property here states that the sampler generates conditionally independent samples but with the same

[2] If we take a 16 x 16 binary image (say). That gives us 1.15 e+77 total number of possibilities of pixel combinations - which is more than the number of stars in the universe!

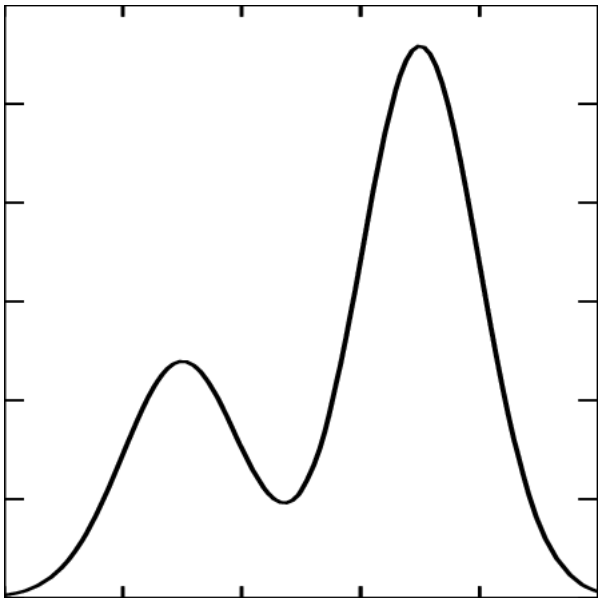


Figure 1.4: An example of an asymmetric multi-modal posterior distribution with the horizontal axis representing the interest variable and the vertical axis the probability

marginal distribution. New samples are generated using a transition operator. Sampling is a very useful tool but is is often the bottleneck for Bayesian methods.

Challenges in sampling

Most MCMC methods rely on *detailed balance* to enforce that the sampling algorithm explores all possible states (*ergodicity*). One significant disadvantage of *detailed balance* is that by definition back tracking of states ensues. This implies that the algorithm visits states that it has previously seen which leads to slow inefficient exploration.

In chapter 2, we take a refreshing approach to this problem by exploring a concept of *generalized detailed balance*. This relaxation helps the algorithm to explore in many cases where it would previously back track. This extension was possible in part due to a renewed understanding of a certain class of sampling methods known as Hamiltonian Monte Carlo [138]. These methods are really efficient at sampling in high dimensional spaces. HMC relies on deterministic dynamics and draws its origins in statistical physics [46]. It is this deterministic dynamics that we employ in our work.

In Chapter 3, we extend the methods previously presented by also trying to simulate wait times using a Markov Jump Process. What this does in effect is to create a transition where the algorithm can wait at the current state in addition to transitioning. Since we look at samples along with their waiting time this helps us sample certain types of distributions faster.

While developing faster sampling methods is an active area of research sampling on

digital circuitry will always remain somewhat slow compared to other types of algorithms for inference. Further, for algorithms such as HMC their only source of error actually stems from discretization errors for their dynamics. That is, in the process of implementing a differential equation numerically the discretization errors introduced affects takes the sampling method away from the distribution of interest.

If we actually employed physical devices for computation we can get around such road blocks quite easily and also benefit from the speed at which these systems might compute. For example, in Chapter 4 we explore the use of analog circuits to sample from certain distributions.

We map the problem of sampling from an interest distribution to that of sampling charge from a capacitor in an Inductance, Capacitance, Resistance (LRC) circuit. The dynamics of charge, the circuit parameters, the circuit elements and the circuit architecture determine the shape of the distribution sampled. We think this is potentially an exciting way to think of computation.

Contributions

Here, I present a list of contributions during my PhD. A portion of these works are presented in this thesis.

- Predicting V1 neural responses to natural movies using the shift invariant bispectrum, Mudigonda, Koster, Stevenson, Hillar, Olshausen, COSYNE 2013 (Appendix B)
- Adaptive Compressed Sensing for Classification, Mudigonda, Joshi, Mueler, Hillar, Sommer, Olshausen, NIPS workshop on high dimensional statistical inference, 2013 (Appendix A)
- Hamiltonian Monte Carlo Without Detailed Balance, Sohl-dickstein, Mudigonda, Deweese, JMLR, ICML, 2014 (Chapter 2)
- A Markov Jump Process for more efficient Hamiltonian Monte Carlo, Berger, Mudigonda, Deweese, Sohl-dickstein, MCQMC, 2016 (Chapter 3)
- Hamitonian Monter Carlo inspired sampling in analog devices, Mudigonda*, Zarcone*, Olshausen, Deweese, Cognitve Computing, 2018 (Chapter 4)
- The HEP. TrkX Project: deep neural networks for HL-LHC online and offline tracking, Farrell, .., Mudigonda, et al. , In EPJ, Web of Conferences 2017
- Segmenting and Tracking Extreme Climate Events using Neural Networks, Mudigonda*, Mahesh*, Kim*, Kashinathan, Kahou, Williams, Michalski, O'Brien, Prabhat, NIPS workshop 2018,

- Exascale Deep Learning for Climate Analytics.. Kurth,, Treichler, Romero, Mudigonda, Luehr., Phillips., Mahesh, Matheson, Deslippe, Fatica, and Houston., arXiv preprint arXiv:1810.01993. Super Computing, 2018

- ClimateNet: Bringing the power of Deep Learning to weather and climate sciences via open datasets and architectures, Mayur Mudigonda*, Kevin Yang*, Jiayi Chen*, Annette Greiner,Karthik Kashinath, Prabhat, Neural Information Processing Systems, 2019, Climate Change: How can AI help?

- Advanced Inner Engineering Experiential Meditation Retreat Improves and Sustains Happiness, Awareness and Positive Mental Health with Associated Activation of Endocannabinoid System and BDNF, (under review) Lancet Psychiatry, Sadhasivam, Alankar, Mathuri, Vishnubotla, Mudigonda, et al. , 2018

- Investigating deep reinforcement learning for grasping objects with an anthropomorphic hand, Mudigonda, Agrawal, Deweese, Malik, Neural Information Processing Systms, Deep RL symposium 2017 (Chapter 5)

- Manipulation by feel: Touch-based control with Deep Predictive Models, Tian, Ebert, Jayaraman, Calandra, Mudigonda, Levine, ICRA 2019 (Chapter 7)

- Object identification using curiosity driven tactile exploration, Mudigonda*, Tickell*, Agrawal, (under preparation Chapter 6)

- Active sensation disrupts correlations in S1 and M1 networks in the mouse neocortex — a sensorimotor account, Mudigonda*, Telian*, Livezey, Zarcone, Adesnik, Deweese, COSYNE 2016 (Chapter 8)

- Modeling temporal structure in S1 M1 cortices in mice, Mudigonda, Telian, Adesnik, Deweese, (under preparation Chapter 8)

Chapter 2

Hamiltonian Monte Carlo Without Detailed Balance

We present a method for performing Hamiltonian Monte Carlo that largely eliminates sample rejection. In situations that would normally lead to rejection, instead a longer trajectory is computed until a new state is reached that can be accepted. This is achieved using Markov chain transitions that satisfy the fixed point equation, but do not satisfy detailed balance. The resulting algorithm significantly suppresses the random walk behavior and wasted function evaluations that are typically the consequence of update rejection. We demonstrate a greater than factor of two improvement in mixing time on three test problems. We release the source code as Python and MATLAB packages.

2.1 Introduction

High dimensional and otherwise computationally expensive probabilistic models are of increasing importance for such diverse tasks as modeling the folding of proteins [169], the structure of natural images [39], or the activity of networks of neurons [19].

Sampling from the described distribution is typically the bottleneck when working with these probabilistic models. Sampling is commonly required when training a probabilistic model, when evaluating the model's performance, when performing inference, and when taking expectations [125]. Therefore, work that improves sampling is fundamentally important.

The most common way to guarantee that a sampling algorithm converges to the correct distribution is via a concept known as detailed balance. Sampling algorithms based on detailed balance are powerful because they allow samples from any target distribution to be generated from almost any proposal distribution, using for instance Metropolis-Hastings acceptance criteria [80]. However, detailed balance also suffers from a critical flaw. By definition forward and reverse transitions occur with equal probability under detailed balance, and samplers that obey detailed balance go backwards exactly as often as they go forwards. The state space is thus explored via a random walk over distances longer than those traversed

by a single draw from the proposal distribution. A random walk only travels a distance $dN^{\frac{1}{2}}$ in N steps, where d is the characteristic step length.

The current state-of-the-art sampling algorithm for probability distributions with continuous state spaces is Hamiltonian Monte Carlo (HMC) [46, 138]. By extending the state space to include auxiliary momentum variables, and then using Hamiltonian dynamics to traverse long iso-probability contours in this extended state space, HMC is able to move long distances in state space in a single update step. However, HMC still relies on detailed balance to accept or reject steps, and as a result still behaves like a random walk – just a random walk with a longer step length. Previous attempts to address this have combined multiple Markov steps that individually satisfy detailed balance into a composite step that does not [85], with limited success [101].

The No-U-Turn Sampler (NUTS) sampling package [83] and the windowed acceptance method of [139] both consider Markov transitions within a set of discrete states generated by repeatedly simulating Hamiltonian dynamics. NUTS generates a set of candidate states around the starting state by running Hamiltonian dynamics forwards and backwards until the trajectory doubles back on itself, or a slice variable constraint is violated. It then chooses a new state at uniform from the candidate states. In windowed acceptance, a transition is proposed between a window of states at the beginning and end of a trajectory, rather than the first state and last state. Within the selected window, a single state is then chosen using Boltzmann weightings. Both NUTS and the windowed acceptance method rely on detailed balance to choose the candidate state from the discrete set.

Here we present a novel discrete representation of the HMC state space and transitions. Using this representation, we derive a method for performing HMC while abandoning detailed balance altogether, by directly satisfying the fixed point equation (or equilibrium condition) restricted to the discrete state space. As a result, random walk behavior in the sampling algorithm is greatly reduced, and the mixing rate of the sampler is substantially improved.

2.2 Sampling

We begin by briefly reviewing some key concepts related to sampling. The goal of a sampling algorithm is to draw characteristic samples $\mathbf{x} \in \mathcal{R}^N$ from a target probability distribution $p(\mathbf{x})$. Without loss of generality, we will assume that $p(\mathbf{x})$ is determined by an energy function $E(\mathbf{x})$,

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left(-E(\mathbf{x})\right). \tag{2.1}$$

Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) [140] is commonly used to sample from probabilistic models. In MCMC a chain of samples is generated by repeatedly drawing new samples $\mathbf{x}'$

from a conditional probability distribution $T(\mathbf{x}'|\mathbf{x})$, where $\mathbf{x}$ is the previous sample. Since $T(\mathbf{x}'|\mathbf{x})$ is a probability density over $\mathbf{x}'$, $\int T(\mathbf{x}'|\mathbf{x})\, d\mathbf{x}' = 1$ and $T(\mathbf{x}'|\mathbf{x}) \geq 0$.

Fixed Point Equation

An MCMC algorithm must satisfy two conditions in order to generate samples from the target distribution $p(\mathbf{x})$. The first is mixing, which requires that repeated application of $T(\mathbf{x}'|\mathbf{x})$ must eventually explore the full state space of $p(\mathbf{x})$. The second condition, sometimes called the equilibrium condition, is that the target distribution $p(\mathbf{x})$ must be a fixed point of $T(\mathbf{x}'|\mathbf{x})$. This second condition can be expressed by the fixed point equation,

$$\int p(\mathbf{x})\, T(\mathbf{x}'|\mathbf{x})\, d\mathbf{x} = p(\mathbf{x}'), \tag{2.2}$$

which requires that when $T(\mathbf{x}'|\mathbf{x})$ acts on $p(\mathbf{x})$, the resulting distribution is unchanged.

Detailed Balance

Detailed balance is the most common way of guaranteeing that the Markov transition distribution $T(\mathbf{x}'|\mathbf{x})$ satisfies the fixed point equation (Equation 2.2). Detailed balance guarantees that if samples are drawn from the equilibrium distribution $p(\mathbf{x})$, then for every pair of states $\mathbf{x}$ and $\mathbf{x}'$ the probability of transitioning from state $\mathbf{x}$ to state $\mathbf{x}'$ is identical to that of transitioning from state $\mathbf{x}'$ to $\mathbf{x}$,

$$p(\mathbf{x})\, T(\mathbf{x}'|\mathbf{x}) = p(\mathbf{x}')\, T(\mathbf{x}|\mathbf{x}'). \tag{2.3}$$

By substitution for $T(\mathbf{x}'|\mathbf{x})$ in the left side of Equation 2.2, it can be seen that if Equation 2.3 is satisfied, then the fixed point equation is also satisfied.

An appealing aspect of detailed balance is that a transition distribution satisfying it can be easily constructed from nearly any proposal distribution, using Metropolis-Hastings acceptance/rejection rules [80]. A primary drawback of detailed balance, and of Metropolis-Hastings, is that the resulting Markov chains always engage in random walk behavior, since by definition detailed balance depends on forward and reverse transitions happening with equal probability.

The primary advance in this paper is demonstrating how HMC sampling can be performed without resorting to detailed balance.

2.3 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) can traverse long distances in state space with single Markov transitions. It does this by extending the state space to include auxiliary momentum variables, and then simulating Hamiltonian dynamics to move long distances along iso-probability contours in the expanded state space.

Extended state space

The state space is extended by the addition of momentum variables $\mathbf{v} \in \mathcal{R}^N$, with identity-covariance Gaussian distribution,

$$p(\mathbf{v}) = (2\pi)^{-\frac{N}{2}} \exp\left(-\frac{1}{2}\mathbf{v}^T\mathbf{v}\right). \tag{2.4}$$

We refer to the combined state space of $\mathbf{x}$ and $\mathbf{v}$ as ζ, such that $\zeta = \{\mathbf{x}, \mathbf{v}\}$. The corresponding joint distribution is

$$p(\zeta) = p(\mathbf{x}, \mathbf{v}) = p(\mathbf{x})\, p(\mathbf{v}) = \frac{(2\pi)^{-\frac{N}{2}}}{Z} \exp\left(-H(\zeta)\right), \tag{2.5}$$

$$H(\zeta) = H(\mathbf{x}, \mathbf{v}) = E(\mathbf{x}) + \frac{1}{2}\mathbf{v}^T\mathbf{v}. \tag{2.6}$$

$H(\zeta)$ has the same form as total energy in a physical system, where $E(\mathbf{x})$ is the potential energy for position $\mathbf{x}$ and $\frac{1}{2}\mathbf{v}^T\mathbf{v}$ is the kinetic energy for momentum $\mathbf{v}$ (mass is set to one).

In HMC samples from $p(\mathbf{x})$ are generated by drawing samples from the joint distribution $p(\mathbf{x}, \mathbf{v})$, and retaining only the $\mathbf{x}$ variables as samples from the desired distribution.

Hamiltonian dynamics

Hamiltonian dynamics govern how physical systems evolve with time. It might be useful to imagine the trajectory of a skateboarder rolling in an empty swimming pool. As she rolls downwards she exchanges potential energy for kinetic energy, and the magnitude of her velocity increases. As she rolls up again she exchanges kinetic energy back for potential energy. In this fashion she is able to traverse long distances across the swimming pool, while at the same time maintaining constant total energy over her entire trajectory.

In HMC, we treat $H(\zeta)$ as the total energy of a physical system, with spatial coordinate $\mathbf{x}$, velocity $\mathbf{v}$, potential energy $E(\mathbf{x})$, and kinetic energy $\frac{1}{2}\mathbf{v}^T\mathbf{v}$. In an identical fashion to the case of the skateboarder in the swimming pool, running Hamiltonian dynamics on this system traverses long distances in $\mathbf{x}$ while maintaining constant total energy $H(\zeta)$. By Equation 2.5, moving along a trajectory with constant energy is identical to moving along a trajectory with constant probability.

Hamiltonian dynamics can be run exactly in reverse by reversing the velocity vector. They also preserve volume in ζ. As we will see, all these properties together mean that Hamiltonian dyamics can be used to propose update steps that move long distances in state space while retaining high acceptance probability.

Operators

The Markov transitions from which HMC is constructed can be understood in terms of several operators acting on ζ. These operators are illustrated in Figure 2.1a. This representation of

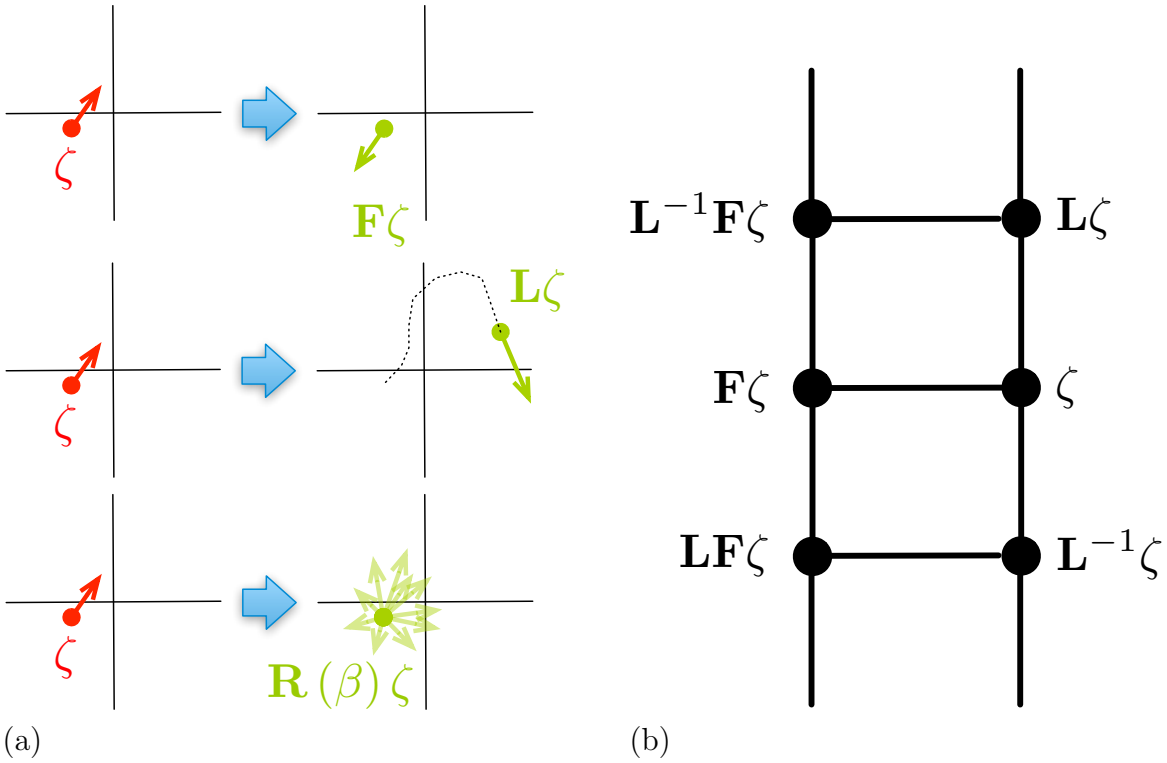


Figure 2.1: *(a)* The action of operators involved in Hamiltonian Monte Carlo (HMC). The base of each red or green arrow represents the position $\mathbf{x}$, and the length and direction of each of these arrows represents the momentum $\mathbf{v}$. The flip operator $\mathbf{F}$ reverses the momentum. The leapfrog operator $\mathbf{L}$ approximately integrates Hamiltonian dynamics. The trajectory taken by $\mathbf{L}$ is indicated by the dotted line. The randomization operator $\mathbf{R}(\beta)$ corrupts the momentum with an amount of noise that depends on β. *(b)* The ladder of discrete states that are accessible by applying $\mathbf{F}$ and $\mathbf{L}$ starting at state ζ. Horizontal movement on the ladder occurs by flipping the momentum, whereas vertical movement occurs by integrating Hamiltonian dynamics.

the actions performed in HMC, and the corresponding state space, is unique to this paper and diverges from the typical presentation of HMC.

Momentum Flip

The momentum flip operator $\mathbf{F}$ reverses the direction of the momentum. It is its own inverse, leaves the total energy unchanged, and preserves volume in state space:

$$\mathbf{F}\zeta = \mathbf{F}\left\{\mathbf{x}, \mathbf{v}\right\} = \left\{\mathbf{x}, -\mathbf{v}\right\}, \tag{2.7}$$

$$\mathbf{F}^{-1}\zeta = \mathbf{F}\zeta, \tag{2.8}$$

$$H\left(\mathbf{F}\zeta\right) = H\left(\zeta\right), \tag{2.9}$$

$$\left|\det\left(\frac{\partial \mathbf{F}\zeta}{\partial \zeta^T}\right)\right| = 1. \tag{2.10}$$

The momentum flip operator $\mathbf{F}$ causes movement between the left and right sides of the state ladder in Figure 2.1b.

Leapfrog Integrator

Leapfrog, or Störmer-Verlet, integration provides a discrete time approximation to Hamiltonian dynamics [76]. The operator $\mathbf{L}\left(\epsilon, M\right)$ performs leapfrog integration for M leapfrog steps with step length ϵ. For conciseness, $\mathbf{L}\left(\epsilon, M\right)$ will be written only as $\mathbf{L}$,

$$\mathbf{L}\zeta = \begin{cases} \text{The state resulting from } M \text{ steps} \\ \text{of leapfrog integration of Hamilto-} \\ \text{nian dynamics with step length } \epsilon. \end{cases} \tag{2.11}$$

Like exact Hamiltonian dynamics, leapfrog dynamics are exactly reversible by reversing the velocity vector, and they also exactly preserve volume in state space. $\mathbf{L}$ can be inverted by reversing the sign of the momentum, tracing out the reverse trajectory, and then reversing the sign of the momentum again so that it points in the original direction;

$$\mathbf{L}^{-1}\zeta = \mathbf{F}\mathbf{L}\mathbf{F}\zeta, \tag{2.12}$$

$$\left|\det\left(\frac{\partial \mathbf{L}\zeta}{\partial \zeta^T}\right)\right| = 1. \tag{2.13}$$

Unlike for exact dynamics, the total energy $H\left(\zeta\right)$ is only approximately conserved by leapfrog integration, and the energy accumulates errors due to discretization. This discretization error in the energy is the source of all rejections of proposed updates in HMC.

The leapfrog operator $\mathbf{L}$ causes movement up the right side of the state ladder in Figure 2.1b, and down the left side of the ladder.

Momentum Randomization

The momentum randomization operator $\mathbf{R}(\beta)$ mixes an amount of Gaussian noise determined by $\beta \in [0,1]$ into the velocity vector,

$$\mathbf{R}(\beta)\,\zeta = \mathbf{R}(\beta)\,\{\mathbf{x},\mathbf{v}\} = \{\mathbf{x},\mathbf{v}'\}, \tag{2.14}$$

$$\mathbf{v}' = \mathbf{v}\sqrt{1-\beta} + \mathbf{n}\sqrt{\beta}, \tag{2.15}$$

$$\mathbf{n} \sim N(\mathbf{0},\mathbf{I}). \tag{2.16}$$

Unlike the previous two operators, the momentum randomization operator is not deterministic. $\mathbf{R}(\beta)$ is however a valid Markov transition operator for $p(\zeta)$ on its own, in that it satisfies both Equation 2.2 and Equation 2.3.

The momentum randomization operator $\mathbf{R}(\beta)$ causes movement off of the current state ladder and onto a new state ladder.

Discrete State Space

As illustrated in Figure 2.1b, the operators $\mathbf{L}$ and $\mathbf{F}$ generate a discrete state space ladder, with transitions only occurring between ζ and three other states. Note that every state on the ladder can be represented many different ways, depending on the series of operators used to reach it. For instance, the state in the upper left of the figure pane can be written $\mathbf{L}^{-1}\mathbf{F}\zeta = \mathbf{FL}\zeta = \mathbf{LFL}\zeta = \cdots$.

Standard HMC can be viewed in terms of transitions on this ladder. Additionally, we will see that this discrete state space view allows Equation 2.2 to be solved directly by replacing the integral over all states with a short sum.

Standard HMC

HMC as typically implemented consists of the following steps. Here, $\zeta^{(t,s)}$ represents the state at sampling step t, and sampling substep s. Each numbered item below corresponds to a valid Markov transition for $p(\zeta)$, satisfying detailed balance. A full sampling step consists of the composition of all three Markov transitions.

1. a) Generate a proposed update,

$$\zeta' = \mathbf{FL}\zeta^{(t,0)}. \tag{2.17}$$

   On the state ladder in Figure 2.1b, this corresponds to moving up one rung ($\mathbf{L}$), and then moving from the right to the left side ($\mathbf{F}$).

   b) Accept or reject the proposed update using Metropolis-Hastings rules,

$$\pi_{accept} = \min\left(1, \frac{p(\zeta')}{p(\zeta)}\right), \tag{2.18}$$

$$\zeta^{(t,1)} = \begin{cases} \zeta' & \text{with probability } \pi_{accept} \\ \zeta^{(t,0)} & \text{with probability } 1-\pi_{accept} \end{cases}. \tag{2.19}$$

Note that since the transition **FL** is its own inverse, the forward and reverse proposal distribution probabilities cancel in the Metropolis-Hastings rule in Equation 3.4.

On rejection, the computations performed in Equation 2.17 are discarded. In our new technique, this will no longer be true.

2. Flip the momentum,

$$\zeta^{(t,2)} = \mathbf{F}\zeta^{(t,1)}. \tag{2.20}$$

If the proposed update from Step 1 was accepted, then this moves $\zeta^{(t,1)}$ from the left back to the right side of the state ladder in Figure 2.1b, and prevents the trajectory from doubling back on itself. If the update was rejected however, and $\zeta^{(t,1)}$ is already on the right side of the ladder, then this causes it to move to the left side of the ladder, and the trajectory to double back on itself.

Doubling back on an already computed trajectory is wasteful in HMC, both because it involves recomputing nearly redundant trajectories, and because the distance traveled before the sampler doubles back is the characteristic length scale beyond which HMC explores the state space by a random walk.

3. Corrupt the momentum with noise,

$$\zeta^{(t+1,0)} = \mathbf{R}\left(\beta\right)\zeta^{(t,2)}. \tag{2.21}$$

It is common to set $\beta = 1$, in which case the momentum is fully randomized every sampling step. In our experiments (Section 2.5) however, we found that smaller values of β produced large improvements in mixing time. This is therefore a hyperparameter that is probably worth adjusting[1].

2.4 Look Ahead HMC

Here we introduce an HMC algorithm that relies on Markov transitions that do not obey detailed balance, but still satisfy the fixed point equation. This algorithm eliminates much of the momentum flipping that occurs on rejection in HMC, and as a result greatly reduces random walk behavior. It also prevents the trajectory computations that would typically be discarded on proposal rejection from being wasted. We call our algorithm Look Ahead Hamiltonian Monte Carlo (LAHMC).

[1] One method for choosing β [39] which we have found to be effective is to set it such that it randomizes a fixed fraction α of the momentum per unit simulation time,

$$\beta = \alpha^{\frac{1}{\epsilon M}}. \tag{2.22}$$

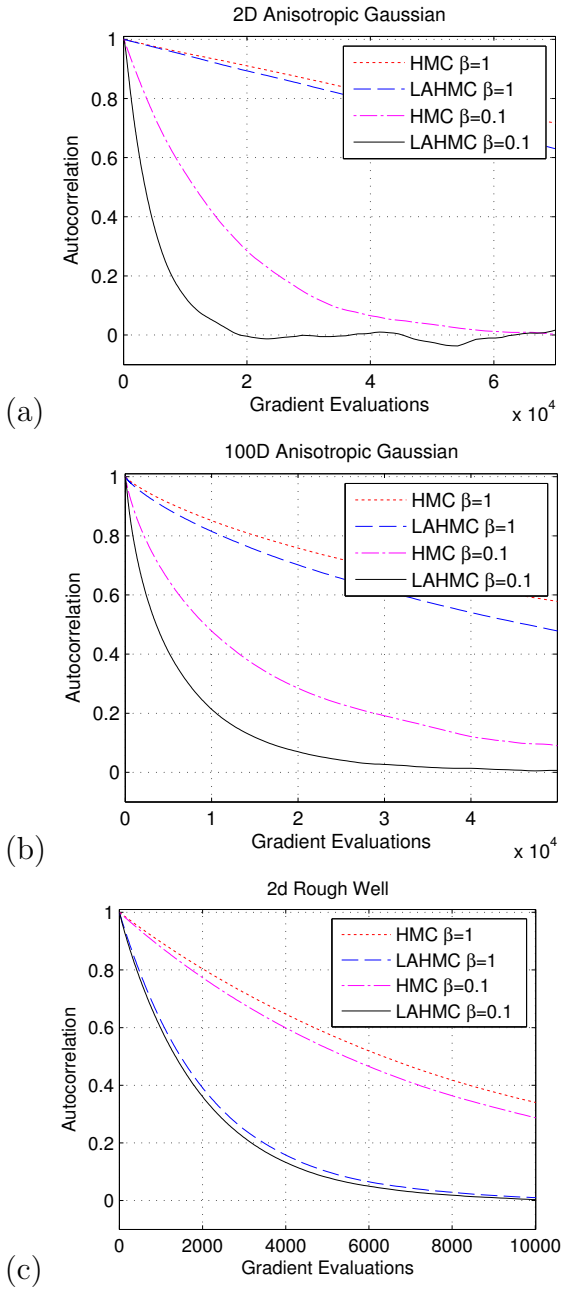


Figure 2.2: Autocorrelation vs. number of function evaluations for standard HMC (no momentum randomization, $\beta = 1$), LAHMC with $\beta = 1$, persistent HMC ($\beta = 0.1$), and persistent LAHMC ($\beta = 0.1$) for *(a)* a two dimensional ill-conditioned Gaussian, *(b)* a one hundred dimensional ill-conditioned Gaussian, and *(c)* a two dimensional well conditioned energy function with a "rough" surface. In all cases the LAHMC sampler demonstrates faster mixing.

Intuition

In LAHMC, in situations that would correspond to a rejection in Step 1 of Section 2.3, we will instead attempt to travel even farther by applying the leapfrog operator $\mathbf{L}$ additional times. This section provides intuition for how this update rule was discovered, and how it can be seen to connect to standard HMC. A more mathematically precise description will follow in the next several sections.

LAHMC can be understood in terms of a series of modifications of standard HMC. The net effect of Steps 1 and 2 in Section 2.3 is to transition from state ζ into either state $\mathbf{L}\zeta$ or state $\mathbf{F}\zeta$, depending on whether the update in Section 2.3 Step 1 was accepted or rejected.

We wish to minimize the transitions into state $\mathbf{F}\zeta$. In LAHMC we do this by replacing as many transitions from ζ to $\mathbf{F}\zeta$ as possible with transitions that instead go from ζ to $\mathbf{L}^2\zeta$. This would seem to change the number of transitions into both state $\mathbf{F}\zeta$ and state $\mathbf{L}^2\zeta$, violating the fixed point equation. However, the changes in incoming transitions from ζ are exactly counteracted because the state $\mathbf{F}\mathbf{L}^2\zeta$ is similarly modified, so that it makes fewer transitions into the state $\mathbf{L}^2\zeta = \mathbf{F}\left(\mathbf{F}\mathbf{L}^2\zeta\right)$, and more transitions into the state $\mathbf{F}\zeta = \mathbf{L}^2\left(\mathbf{F}\mathbf{L}^2\zeta\right)$.

For some states, after this modification there will still be transitions between the states ζ and $\mathbf{F}\zeta$. In order to further minimize these transitions, the process in the preceding paragraph is repeated for these remaining transitions and the state $\mathbf{L}^3\zeta$. This process is then repeated again for states $\mathbf{L}^4\zeta$, $\mathbf{L}^5\zeta$, etc, up to some maximum number of leapfrog applications K.

Algorithm

LAHMC consists of the following two steps,

1. Transition to a new state by applying the leapfrog operator $\mathbf{L}$ between 1 and $K \in \mathcal{Z}^+$ times, or by applying the momentum flip operator $\mathbf{F}$,

$$\zeta^{(t,1)} = \begin{cases} \mathbf{L}\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{L}^1}\left(\zeta^{(t,0)}\right) \\ \mathbf{L}^2\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{L}^2}\left(\zeta^{(t,0)}\right) \\ \cdots & \\ \mathbf{L}^K\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{L}^K}\left(\zeta^{(t,0)}\right) \\ \mathbf{F}\zeta^{(t,0)} & \text{with probability } \pi_{\mathbf{F}}\left(\zeta^{(t,0)}\right) \end{cases}. \tag{2.23}$$

   Note that there is no longer a Metropolis-Hastings accept/reject step. The state update in Equation 2.23 is a valid Markov transition for $p\left(\zeta\right)$ on its own.

2. Corrupt the momentum with noise in an identical fashion as in Equation 2.21,

$$\zeta^{(t+1,0)} = \mathbf{R}\left(\beta\right)\zeta^{(t,1)}. \tag{2.24}$$

Transition Probabilities

We choose the probabilities $\pi_{\mathbf{L}^a}(\zeta)$ for the leapfrog transitions from state ζ to state $\mathbf{L}^a\zeta$ to be

$$\pi_{\mathbf{L}^a}(\zeta) = \min\left[1 - \sum_{b<a} \pi_{\mathbf{L}^b}(\zeta), \frac{p(\mathbf{F}\mathbf{L}^a\zeta)}{p(\zeta)}\left(1 - \sum_{b<a} \pi_{\mathbf{L}^b}(\mathbf{F}\mathbf{L}^a\zeta)\right)\right]. \tag{2.25}$$

Equation 2.25 greedily sets the transition probability $\pi_{\mathbf{L}^a}(\zeta)$ as large as possible, subject to the restrictions that the total transition probability out of state ζ not exceed 1, and that the transition rate in the forward direction ($\zeta \to \mathbf{L}^a\zeta$) not exceed the transition rate in the reverse direction ($\mathbf{F}\mathbf{L}^a\zeta \to \mathbf{F}\zeta$)[2].

Any remaining unassigned probability is assigned to the momentum flip transition,

$$\pi_{\mathbf{F}}(\zeta) = 1 - \sum_a \pi_{\mathbf{L}^a}(\zeta). \tag{2.27}$$

Note that transitions will be performed in a greedy fashion. It is only necessary to compute the state $\mathbf{L}^a\zeta$ and the transition probability $\pi_{\mathbf{L}^a}(\zeta)$ if none of the transitions to states $\mathbf{L}^b\zeta$, for $b < a$, have been taken.

Fixed Point Equation

We can substitute the transition rates from Section 2.4 into the left side of Equation 2.2, and verify that they satisfy the fixed point equation. Note that the integral over all states is transformed into a sum over all source states from which transitions into state ζ might be

[2] Although these transition probabilities do not satisfy detailed balance, as observed in [24] they do satisfy an alternate condition common in physics which is known as *generalized* detailed balance. This follows from the observation that

$$p(\zeta)\,\pi_{\mathbf{L}^a}(\zeta) = p(\mathbf{F}\mathbf{L}^a\zeta)\,\pi_{\mathbf{L}^a}(\mathbf{F}\mathbf{L}^a\zeta). \tag{2.26}$$

Distribution	Sampler	$\mathbf{F}\zeta$	$\mathbf{L}\zeta$	$\mathbf{L}^2\zeta$	$\mathbf{L}^3\zeta$	$\mathbf{L}^4\zeta$
2d Gaussian	HMC $\beta = 1$	0.079	0.921	0	0	0
2d Gaussian	LAHMC $\beta = 1$	0.000	0.921	0.035	0.044	0.000
2d Gaussian	HMC $\beta = 0.1$	0.080	0.920	0	0	0
2d Gaussian	LAHMC $\beta = 0.1$	0.000	0.921	0.035	0.044	0.000
100d Gaussian	HMC $\beta = 1$	0.147	0.853	0	0	0
100d Gaussian	LAHMC $\beta = 1$	0.047	0.852	0.059	0.035	0.006
100d Gaussian	HMC $\beta = 0.1$	0.147	0.853	0	0	0
100d Gaussian	LAHMC $\beta = 0.1$	0.047	0.852	0.059	0.035	0.006
2d Rough Well	HMC $\beta = 1$	0.446	0.554	0	0	0
2d Rough Well	LAHMC $\beta = 1$	0.292	0.554	0.099	0.036	0.019
2d Rough Well	HMC $\beta = 0.1$	0.446	0.554	0	0	0
2d Rough Well	LAHMC $\beta = 0.1$	0.292	0.554	0.100	0.036	0.019

Table 2.1: A table showing the fraction of transitions which occurred to each target state for the conditions plotted in Figure 2.2. Note that LAHMC has far fewer momentum flips than standard HMC.

initiated.

$$
\begin{aligned}
&\int d\zeta' p(\zeta') T(\zeta|\zeta') \\
&\quad = \int d\zeta' p(\zeta') \Bigg(\sum_a \pi_{\mathbf{L}^a}(\zeta') \delta(\zeta - \mathbf{L}^a \zeta') \\
&\qquad\qquad + \pi_{\mathbf{F}}(\zeta') \delta(\zeta - \mathbf{F}\zeta') \Bigg), && (2.28) \\
&\quad = \sum_a p\left(\mathbf{L}^{-a}\zeta\right) \pi_{\mathbf{L}^a}\left(\mathbf{L}^{-a}\zeta\right) + p\left(\mathbf{F}^{-1}\zeta\right) \pi_{\mathbf{F}}\left(\mathbf{F}^{-1}\zeta\right), && (2.29) \\
&\quad = \sum_a p(\mathbf{F}\mathbf{L}^a\mathbf{F}\zeta) \pi_{\mathbf{L}^a}(\mathbf{F}\mathbf{L}^a\mathbf{F}\zeta) + p(\mathbf{F}\zeta) \pi_{\mathbf{F}}(\mathbf{F}\zeta), && (2.30) \\
&\quad = \sum_a p(\mathbf{F}\zeta) \pi_{\mathbf{L}^a}(\mathbf{F}\zeta) + p(\mathbf{F}\zeta) \pi_{\mathbf{F}}(\mathbf{F}\zeta), && (2.31) \\
&\quad = p(\mathbf{F}\zeta) \left[\sum_a \pi_{\mathbf{L}^a}(\mathbf{F}\zeta) + \pi_{\mathbf{F}}(\mathbf{F}\zeta) \right], && (2.32) \\
&\quad = p(\zeta). && (2.33)
\end{aligned}
$$

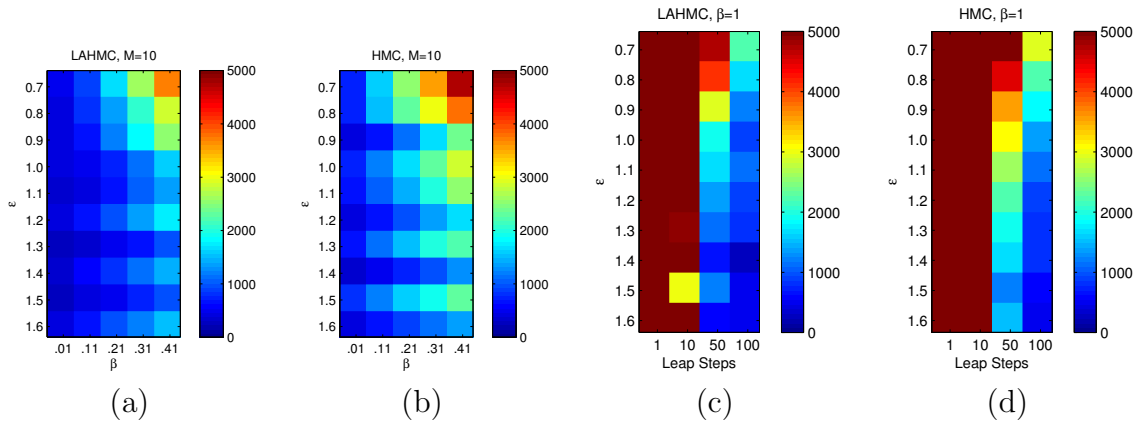


Figure 2.3: Images illustrating mixing time as a function of HMC hyperparameters for a two dimensional ill-conditioned Gaussian distribution. Pixel intensity indicates the number of gradient evaluations required to reach an autocorrelation of 0.5. LAHMC always outperforms HMC for the same hyperparameter settings. *(a)* LAHMC as a function of ϵ and β, for fixed $M = 10$, *(b)* HMC as a function of ϵ and β, for fixed $M = 10$, *(c)* LAHMC as a function of ϵ and M, for fixed $\beta = 1$, *(d)* HMC as a function of ϵ and M, for fixed $\beta = 1$.

2.5 Experimental Results

As illustrated in Figure 2.2, we compare the mixing time for our technique and standard HMC on three distributions. HMC and LAHMC both had step length and number of leapfrog steps set to $\epsilon = 1$, and $M = 10$. Values of β were set to 1 or 0.1 as stated in the legend. For LAHMC the maximum number of leapfrog applications was set to $K = 4$. In all cases, LAHMC outperformed standard HMC for the same setting of hyperparameters, often by more than a factor of 2.

The first two target distributions are 2 and 100 dimensional ill-conditioned Gaussian distributions. In both Gaussians, the eigenvalues of the covariance matrix are log-linearly distributed between 1 and 10^6.

The final target distribution was chosen to demonstrate that LAHMC is useful even for well conditioned distributions. The energy function used was the sum of an isotropic quadratic and sinusoids in each of two dimensions,

$$E(\mathbf{x}) = \frac{1}{2\sigma_1^2}\left(x_1^2 + x_2^2\right) + \cos\left(\frac{\pi x_1}{\sigma_2}\right) + \cos\left(\frac{\pi x_2}{\sigma_2}\right), \tag{2.34}$$

where $\sigma_1 = 100$ and $\sigma_2 = 2$. Although this distribution is well conditioned the sinusoids cause it to have a "rough" surface, such that traversing the quadratic well while maintaining a reasonable discretization error requires many leapfrog steps.

The fraction of the sampling steps resulting in each possible update for the samplers and energy functions in Figure 2.2 is illustrated in Table 2.1. The majority of momentum flips in

standard HMC were eliminated by LAHMC. Note that the acceptance rate for HMC with these hyperparameter values is reasonably close to its optimal value of 65% [138].

Figure 2.3 shows several grid searches over hyperparameters for a two dimensional ill-conditioned Gaussian, and demonstrates that our technique outperforms standard HMC for all explored hyperparameter settings. Due to computational constraints, the eigenvalues of the covariance of the Gaussian are 1 and 10^5 in Figure 2.3, rather than 1 and 10^6 as in Figure 2.2a.

MATLAB and Python implementations of LAHMC are available at `http://github.com/Sohl-Dickstein/LAHMC`. Figure 2.2 and Table 2.1 can be reproduced by running generate_figure_2.m or generate_figure_2.py.

2.6 Future Directions

There are many powerful variations on standard HMC that are complementary to and could be combined naturally with the present work. These include Riemann manifold HMC [64], quasi-Newton HMC [202], Hilbert space HMC [13], shadow Hamiltonian methods [91], parameter adaptation techniques [191], Hamiltonian annealed importance sampling [179], split HMC [170], and tempered trajectories [138].

It should be possible to further reduce random walk behavior by exploring new topologies and allowed state transitions. Two other schemes have already been explored, though with only marginal benefit. In one scheme as many flips as possible are replaced by identity transitions. This is described in the note [177]. In a second scheme, a state space is constructed with two sets of auxiliary momentum variables, and an additional momentum-swap operator which switches the two momenta with each other is included in the allowed transitions. In this scenario, in situations that would typically lead to momentum flipping, with high probability the two sets of momenta can instead be exchanged with each other. This leads to momentum randomization on rejection, rather than momentum reversal. Unfortunately, though this slightly improves mixing time, it still amounts to a random walk on a similar length scale. The exploration of other topologies and allowed transitions will likely prove fruitful.

Any deterministic, reversible, discrete stepped trajectory through a state space can be mapped onto the ladder structure in Figure 2.1. The Markov transition rules presented in this paper could therefore be applied to a wide range of problems. All that is required in addition to the mapping is an auxiliary variable indicating direction along that trajectory. In HMC, the momentum variable doubles as a direction indicator, but there could just as easily be an additional variable $d \in \{-1, 1\}$, $p(d = 1) = \frac{1}{2}$, which indicates whether transitions are occurring up or down the ladder. The efficiency of the exploration then depends only on choosing a sensible, approximately energy conserving, trajectory.

2.7 Acknowledgements

The work in this chapter was performed in collaboration with Jascha Sohl-Dickstein with advise from Bruno Olshausen and Mike Deweese. The work was presented at the International Conference on Machine Learning, 2014, Beijing, China and also featured as an article in the Journal of Machine Learning Research, and was presented at the MCQMC conference held at Stanford in Aug 2016.

Chapter 3

A Markov Jump Process for More Efficient Hamiltonian Monte Carlo

In most sampling algorithms, including Hamiltonian Monte Carlo, transition rates between states correspond to the probability of making a transition in a single time step, and are constrained to be less than or equal to 1. We derive a Hamiltonian Monte Carlo algorithm using a continuous time Markov jump process, and are thus able to escape this constraint. Transition rates in a Markov jump process need only be non-negative. We demonstrate that the new algorithm leads to improved mixing for several example problems, both by evaluating the spectral gap of the Markov operator, and by computing autocorrelation as a function of compute time. We release the algorithm as an open source Python package.

3.1 Introduction

Efficient sampling is a challenge in many tasks involving high dimensional probabilistic models, in a diversity of fields. For example, sampling is commonly required to train a probabilistic model, to evaluate the model's performance, to perform inference, and to take expectations under the model [125].

In this paper we introduce a method for more efficient sampling, by making Markov transitions in continuous rather than discrete time. This allows transitions which have lower probability in discrete time Monte Carlo to occur more often, with a shorter time spent for each visit. It thus allows more rapid exploration of state space. We apply this approach to develop a novel Hamiltonian Monte Carlo (HMC) sampling algorithm. Finally, we demonstrate the effectiveness of this approach by comparing both spectral gaps and autocorrelation on several example problems.

Discrete time sampling

Most sampling algorithms involve transitioning between states in discrete time steps. In this discrete-time framework, the transition rates out of a state must sum to 1 and be non-negative. In fact, the popular Metropolis-Hastings acceptance rule [80] for Markov Chain Monte Carlo (MCMC) works well because it maximizes the transition rate between a pair of states, subject to this constraint and to detailed balance. As we will see, however, this constraint on transition rates limits performance, and better mixing can be achieved by allowing transition rates larger than 1.

Markov jump process

A Markov process can instead be expressed in continuous time, in which case the only restriction on the transition rates between distinct states is that they be non-negative. In continuous time, the rate of transition from a state j into a state $i \neq j$ is given by Γ_{ij}, and the rate of change of the probability p_i of state i is

$$\frac{\partial p_i}{\partial t} = \sum_j \Gamma_{ij} p_j, \tag{3.1}$$

where $\Gamma_{ij} \geq 0$ for $\forall i \neq j$, and we use the convention $\Gamma_{jj} = -\sum_{i \neq j} \Gamma_{ij}$.

A particle evolving in a system of this form makes stochastic transitions between a set of discrete states in continuous time. Each transition is governed by a Poisson process. Neglecting other states, the waiting time w_{ij} for a transition from state j to state i will be drawn from an exponential distribution, $P(w_{ij}) = \Gamma_{ij} \exp(-\Gamma_{ij} w_{ij})$.

To simulate the system, waiting times w_{kj} are generated for all candidate states k, and the shortest waiting time (with index $i = \text{argmin}_k w_{kj}$) is chosen. This shortest waiting time is called the holding time. A transition is then performed to state i after a delay of length w_{ij}.

A system that evolves in this way is known as a Markov jump process [35].

Markov jump processes and sampling

A large body of work exists using Markov jump processes to generate samples for physical systems, such as chemical reactions, which are well described by stochastic transitions between discrete states. These algorithms are often referred to as kinetic Monte Carlo [187], dynamical Monte Carlo [50], the Gillespie algorithm [63], or directly as Markov jump processes [6]. Additional work has addressed efficient sampling of trajectories in Markov jump processes [155] and the statistical properties of these trajectories [133].

To our knowledge, Markov jump processes have not previously been applied to general purpose Monte Carlo sampling. That is, they have not been used to sample from arbitrary probability distributions which lack a natural interpretation as resulting from temporal dynamics. However, see [72] where a Markov jump process is used to sample from a posterior distribution over model graph structure.

Markov jump processes have properties which might be expected to accelerate general purpose Monte Carlo. They visit low probability states far more frequently than a typical discrete time sampler, while remaining in those states for only a short time period (short holding time). This has the potential to lead to faster mixing, in a similar fashion to parallel tempering [128] or annealing. Unlike annealing the sampling chain in Markov jump HMC always has the correct target distribution as a fixed point. Unlike parallel tempering only a single sampling chain need be run. Markov jump processes additionally have no self-transitions, and thus eliminate the rejection step typically found in discrete time Monte Carlo.

However, unlike in discrete time Monte Carlo, in a Markov jump process it is necessary to compute transition rates to all possible target states i from the current state j (though see [197, 196]). This can be prohibitively expensive. As we will see, for HMC we will only need to consider a small number of target states, making Markov jump processes computationally efficient. **The primary contribution of this paper is to use a Markov jump process to develop a faster HMC algorithm**.

Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) [46, 138] is the state-of-the-art, general purpose Monte Carlo algorithm for sampling from a distribution $\pi(\mathbf{x})$ over a continuous state space $\mathbf{x} \in \mathbb{R}^N$.

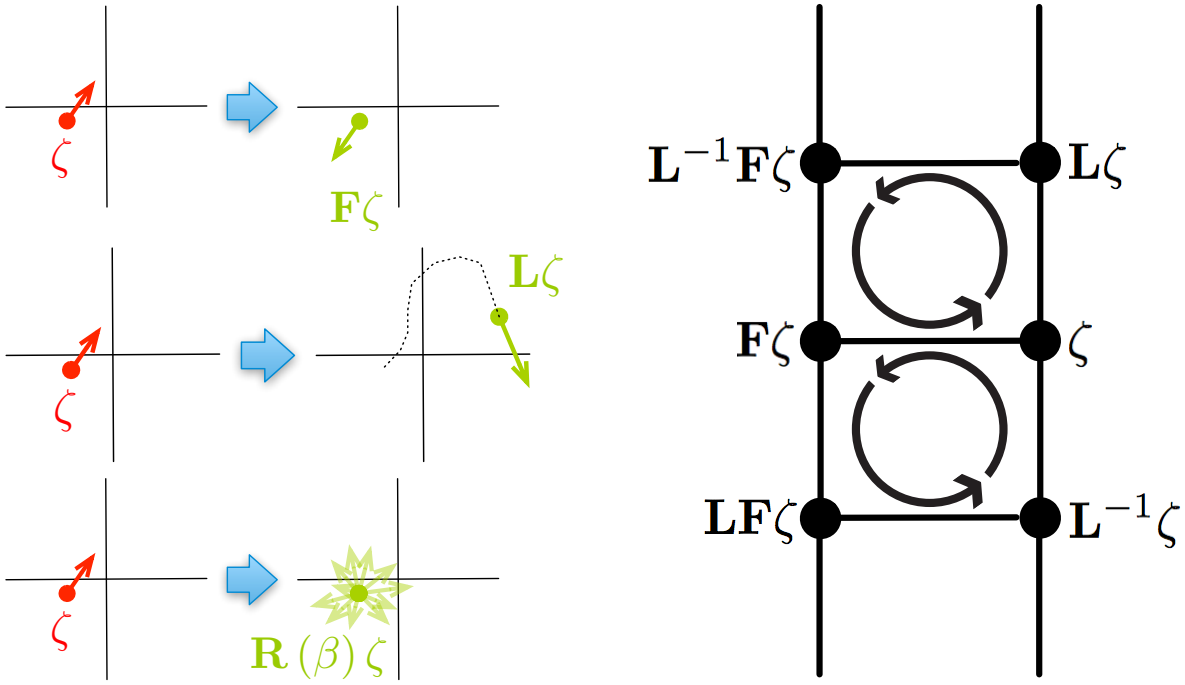
HMC utilizes the same Hamiltonian dynamics that govern the evolution of a physical system – for instance a marble rolling in a swimming pool – to rapidly traverse long distances in state space. In HMC the state space is first extended to include auxiliary momentum variables $\mathbf{v} \in \mathbb{R}^N$ with distribution $\pi(\mathbf{v})$, such that the joint state space over position and momentum is $\boldsymbol{\zeta} = \{\mathbf{x}, \mathbf{v}\}$, with joint distribution $\pi(\boldsymbol{\zeta}) = \pi(\mathbf{x})\pi(\mathbf{v})$. An analogy is then made between $\mathbf{x}$ and physical position (e.g. the position of the marble), between $\mathbf{v}$ and physical momentum (the momentum of the marble), and between $(-\log \pi(\mathbf{x}))$ and potential energy (the height of the swimming pool at position $\mathbf{x}$). Since physical dynamics conserve energy, they can generate very long trajectories in state space while remaining on a constant probability contour of $\pi(\boldsymbol{\zeta})$.

HMC is thus able to move very long distances in state space in a single update step.

Discrete state space Ladder

As introduced in [180] and illustrated in Figure 3.1, HMC can be viewed in terms of transitions on a discrete state space ladder. This state ladder is formulated by expressing the action of HMC on a sampling particle in terms of three operators. The leapfrog integration operator, $\mathbf{L}$, approximately integrates Hamiltonian dynamics for a fixed number of time steps and a fixed step length.

The momentum flip operator, $\mathbf{F}$, reverses a particle's direction of travel along a contour by flipping its momentum. The momentum randomization operator $\mathbf{R}$ redraws the momentum vector from $\pi(\mathbf{v})$, and moves a particle onto a new state space ladder.



(a) (b)

Figure 3.1: *(a)* The action of operators involved in Hamiltonian Monte Carlo (HMC). The base of each red or green arrow represents the position $\mathbf{x}$, and the length and direction of each of these arrows represents the momentum $\mathbf{v}$. The flip operator $\mathbf{F}$ reverses the momentum. The leapfrog operator $\mathbf{L}$ approximately integrates Hamiltonian dynamics. The trajectory taken by $\mathbf{L}$ is indicated by the dotted line. The randomization operator $\mathbf{R}$ replaces the momentum. *(b)* The ladder of discrete states generated by the leapfrog ($\mathbf{L}$) and flip ($\mathbf{F}$) operators. Application of $\mathbf{F}$ corresponds to movement across the 'rungs' of the ladder. Application of $\mathbf{L}$ corresponds to movement up the right side of the ladder, or down the left. Inset arrows illustrate closed loops of constant total probability flow under our chosen rate (Equation 3.4)

This perspective suggests a powerful formalism, effectively discretizing the state space and illuminating the structure[1] of HMC. Throughout this work, we refer to the structure generated by the operators as a state ladder. As illustrated in Figure 3.1, $\mathbf{L}$ causes movement up the right side of a state ladder and down the left side, whereas $\mathbf{F}$ causes horizontal movement across the rungs of a ladder. A trajectory can be exactly reversed by reversing the momentum, integrating Hamiltonian dynamics, and reversing the momentum again. As can be seen in Figure 3.1, this corresponds to making a loop on the state space ladder, and it implies that $\mathbf{FLFL} = \mathbf{I}$, where $\mathbf{I}$ is the identity operator. $\mathbf{R}$ causes movement off of the current state ladder and onto a new one. Both $\mathbf{F}$ and $\mathbf{L}$ are volume preserving, which will eliminate the need to consider the determinant of the Jacobian (which captures volume changes) when computing Markov transition rates.

If we only allow transitions between states that are connected on the state ladder (Figure 3.1), then transitions can only occur between $\boldsymbol{\zeta}$ and three other states ($\mathbf{L}^{-1}\boldsymbol{\zeta}$, $\mathbf{F}\boldsymbol{\zeta}$, $\mathbf{L}\boldsymbol{\zeta}$). This makes HMC well matched to Markov jump processes, since only a small number of transitions need be considered.

Current research

Improvements and elaborations on HMC are an area of extremely active research. Although our approach is novel, other approaches to improving the utility of HMC include the use of shadow Hamiltonians that are more closely conserved by the approximate Hamiltonian integrator [91], Riemann manifold HMC [64] and other investigation of its geometry [14], quasi-Newton HMC [202], Hilbert space HMC [13], parameter adaptation techniques [191], Hamiltonian annealed importance sampling [179], split HMC [170], tempered trajectories [138], novel discrete time transition rules [180, 177, 23], stochastic gradient variants on HMC [32], HMC for approximate Bayesian computation [132], and new approximate Hamiltonian integrators [30].

3.2 Markov jump Hamiltonian Monte Carlo

We now develop the MJHMC sampler, schematized in Figure 3.2. Sections 3.2 through 3.2 develop the Markov transition rates. The correctness of the Markov transition rates are verified in Appendix 3.5. Section 3.2 establishes the connection between simulation time and compute cost. Section 3.2 provides pseudocode for the algorithm.

[1] Neglecting momentum randomization, HMC acts in a manner isomorphic to the Dihedral group of, in general, infinite order. The HMC state ladder, and thus the order, is generally infinite because trajectories through state space produced by Hamiltonian dynamics are almost never closed.

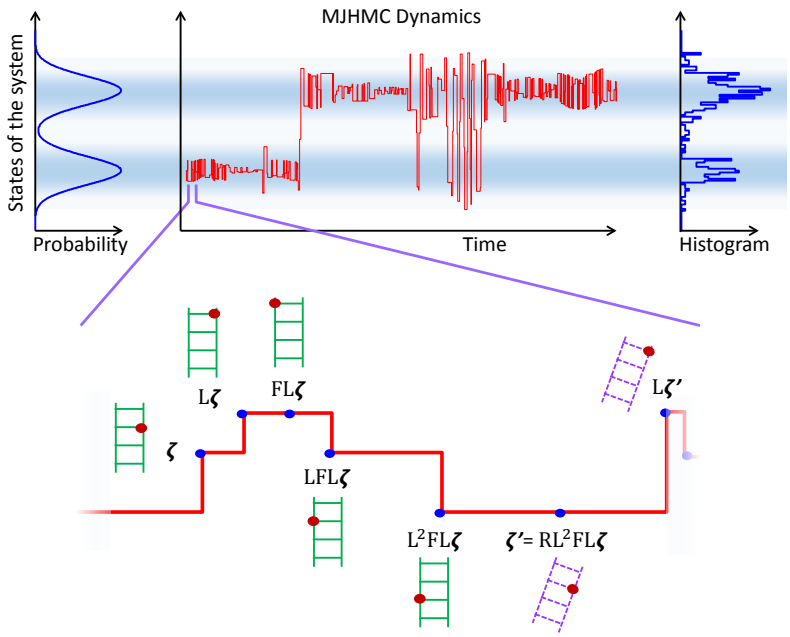


Figure 3.2: Illustration of Markov Jump HMC sampling dynamics. The red curve represents a particle trajectory for 400 time steps. Blue shading indicates the probability density, plotted at the left, and an empirical histogram of the samples is shown at the right. The inset blowup at the bottom shows how movement of the sampling particle corresponds to transitions on the state ladder, using the symbolic and graphical conventions described in Figure 3.1. Note that the sampling particle dwells in a position for a duration related to the probability density of that state relative to its neighbors. We have indicated the transition from one state ladder (vertical green ladder) to a new ladder (angled purple ladder) following a momentum randomization event, resulting in a new state labeled $\boldsymbol{\zeta}'$.

Continuous time transition rates on HMC state ladder

A Markov process must satisfy *two conditions* [100] to sample from a target distribution . The first is ergodicity, which requires that the process will eventually explore the full state space; this is typically straightforward to satisfy.

The second condition is that the target distribution must be a fixed point of the Markov process. This is usually achieved via dynamics that satisfy detailed balance matched to the state probabilities of the target distribution.

Closed loops preserve the fixed point distribution

Markov transition rates $\tilde{\Gamma}(\boldsymbol{\zeta}' \mid \boldsymbol{\zeta})$ that preserve $\pi(\boldsymbol{\zeta})$ as a fixed point can be constructed from closed loops of constant probability flow in state space.

For closed loops to have constant probability flow, the flow $r(\boldsymbol{\zeta}_i \mid \boldsymbol{\zeta}_j) = \Gamma(\boldsymbol{\zeta}_i \mid \boldsymbol{\zeta}_j)\pi(\boldsymbol{\zeta}_j)$

from state j to i must be identical for each link in the loop. This is analogous to Kirchoff's current law for electrical circuits – constant probability flow in all loops implies that the net probability flow into any state equals the net flow out of that state.

Choosing transition rates

We set the loops to be between the "rungs" of the state ladder, as illustrated in Figure 3.1. The loop balance condition for each closed loop becomes

$$\tilde{r}(\mathbf{F}\boldsymbol{\zeta} \mid \boldsymbol{\zeta}) = \tilde{r}(\mathbf{LF}\boldsymbol{\zeta} \mid \mathbf{F}\boldsymbol{\zeta}) = \tilde{r}(\mathbf{L^{-1}}\boldsymbol{\zeta} \mid \mathbf{LF}\boldsymbol{\zeta}) = \tilde{r}(\boldsymbol{\zeta} \mid \mathbf{L}^{-1}\boldsymbol{\zeta}), \tag{3.2}$$

$$\begin{aligned} \pi(\boldsymbol{\zeta})\,\tilde{\Gamma}(\mathbf{F}\boldsymbol{\zeta} \mid \boldsymbol{\zeta}) &= \pi(\mathbf{F}\boldsymbol{\zeta})\,\tilde{\Gamma}(\mathbf{LF}\boldsymbol{\zeta} \mid \mathbf{F}\boldsymbol{\zeta}) \\ &= \pi(\mathbf{LF}\boldsymbol{\zeta})\,\tilde{\Gamma}\left(\mathbf{L^{-1}}\boldsymbol{\zeta} \mid \mathbf{LF}\boldsymbol{\zeta}\right) \\ &= \pi\left(\mathbf{L^{-1}}\boldsymbol{\zeta}\right)\tilde{\Gamma}\left(\boldsymbol{\zeta} \mid \mathbf{L}^{-1}\boldsymbol{\zeta}\right). \end{aligned} \tag{3.3}$$

In order to satisfy these conditions, we set the transition rates to be

$$\tilde{\Gamma}(\boldsymbol{\zeta}' \mid \boldsymbol{\zeta}) = \begin{cases} \left[\frac{\pi(\mathbf{L}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} & \boldsymbol{\zeta}' = \mathbf{L}\boldsymbol{\zeta} \\ \left[\frac{\pi(\mathbf{LF}\boldsymbol{\zeta})}{\pi(\mathbf{F}\boldsymbol{\zeta})}\right]^{\frac{1}{2}} & \boldsymbol{\zeta}' = \mathbf{F}\boldsymbol{\zeta} \\ 0 & \text{otherwise} \end{cases}. \tag{3.4}$$

One can verify by direct substitution that these transition rates satisfy Equations 3.2 and 3.3. The transition rates for the full ladder consist of a sum over the transition rates for each loop.

Opposing flows cancel

As can be seen in Figure 3.1, adjacent loops make flip transitions across the "rungs" of the ladder in opposite directions. After summing over all loops, the net transitions across the ladder approximately cancel.

This allows us to reduce the flip rates in both directions, such that the flip rate in one direction is zero. The final rate of flip transitions in our algorithm will thus be

$$\Gamma(\mathbf{F}\boldsymbol{\zeta} \mid \boldsymbol{\zeta}) = \tilde{\Gamma}(\mathbf{F}\boldsymbol{\zeta} \mid \boldsymbol{\zeta}) - \min\left[\tilde{\Gamma}(\mathbf{F}\boldsymbol{\zeta} \mid \boldsymbol{\zeta}), \tilde{\Gamma}(\boldsymbol{\zeta} \mid \mathbf{F}\boldsymbol{\zeta})\right], \tag{3.5}$$

$$= \max\left[0, \tilde{\Gamma}(\mathbf{F}\boldsymbol{\zeta} \mid \boldsymbol{\zeta}) - \tilde{\Gamma}(\boldsymbol{\zeta} \mid \mathbf{F}\boldsymbol{\zeta})\right], \tag{3.6}$$

$$= \max\left[0, \left[\frac{\pi(\mathbf{LF}\boldsymbol{\zeta})}{\pi(\mathbf{F}\boldsymbol{\zeta})}\right]^{\frac{1}{2}} - \left[\frac{\pi(\mathbf{LFF}\boldsymbol{\zeta})}{\pi(\mathbf{FF}\boldsymbol{\zeta})}\right]^{\frac{1}{2}}\right], \tag{3.7}$$

$$= \max\left[0, \left[\frac{\pi(\mathbf{L}^{-1}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} - \left[\frac{\pi(\mathbf{L}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}}\right], \tag{3.8}$$

where the final step relies on the observations that $\mathbf{F}\mathbf{F}\boldsymbol{\zeta} = \boldsymbol{\zeta}$, and that $\pi(\boldsymbol{\zeta}) = \pi(\mathbf{F}\boldsymbol{\zeta})$ [180]. In practice, $\pi(\mathbf{L}^{-1}\boldsymbol{\zeta})$ will typically already be available (up to a shared normalization constant) from the preceding Markov transition, and will not need to be computed.

Due to discretization error, the leapfrog integrator for Hamiltonian dynamics only approximately conserves probability. Equation 3.8 shows that the residual flow across the ladder stems from this discretization error of the leapfrog integrator. This is completely analogous to the cause of momentum flips in standard HMC.

Momentum randomization

In discrete time HMC the momentum is periodically corrupted with noise. If this was not done, then sampling would be restricted to a single state ladder, and mixing between state ladders would not occur. In order to accomplish the same end in continuous time, we jump to a state $\mathbf{R}\boldsymbol{\zeta}$ with a constant transition rate β. A transition to $\mathbf{R}\boldsymbol{\zeta}$ corresponds to replacing the momentum $\mathbf{v}$ with a new draw from π. The transition rate from $\mathbf{v}$ to a particular $\mathbf{v}'$ is thus $\beta\pi(\mathbf{v}')$. It can be seen by substitution that the momentum randomization kernel, on its own, satisfies detailed balance and thus leaves the target distribution as a fixed point of the dynamics.

Final transition rates

Combining the transition rates derived in Sections 3.2, 3.2, and 3.2,

$$\Gamma(\boldsymbol{\zeta}' \mid \boldsymbol{\zeta}) = \begin{cases} \left[\frac{\pi(\mathbf{L}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} & \boldsymbol{\zeta}' = \mathbf{L}\boldsymbol{\zeta} \\ \max\left[0, \left[\frac{\pi(\mathbf{L}^{-1}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} - \left[\frac{\pi(\mathbf{L}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}}\right] & \boldsymbol{\zeta}' = \mathbf{F}\boldsymbol{\zeta} \\ \beta & \boldsymbol{\zeta}' = \mathbf{R}\boldsymbol{\zeta} \\ 0 & \text{otherwise} \end{cases} . \tag{3.9}$$

We verify that these transition rates satisfy the balance condition for $\pi(\boldsymbol{\zeta})$ in Appendix 3.5. In the third line $\mathbf{R}\boldsymbol{\zeta}$ does not correspond to a single fixed state, but rather indicates that the momentum is replaced by a new draw from $\pi(\mathbf{v})$, where this replacement is triggered by a Poisson process with rate β. Note that as in [180] the overall dynamics do *not* satisfy detailed balance, and can be expected to mix more quickly as a result [89].

System time vs compute time

We have described continuous time dynamics in terms of a system, or simulation, time. However, when applying this sampler to a real problem it is its performance as measured relative to compute time that matters. Here we show how to relate the continuous time dynamics of the Markov jump process to a discrete time Markov process, with an approximately fixed computational cost per time-step.

First we observe that there is a discrete time Markov process describing only the sequence of visited states, thus neglecting the holding time spent in each state. For notational convenience we represent Markov processes using matrix notation in this section. The update rule for this Markov process can be written

$$\hat{\mathbf{p}}^{\tau+1} = \hat{\mathbf{T}}\hat{\mathbf{p}}^{\tau}, \tag{3.10}$$

$$\hat{T}_{ij} = \begin{cases} \prod_{k \neq j} \frac{\Gamma_{ij}}{\Gamma ij + \Gamma ik} & i \neq j \\ 0 & i = j \end{cases} \tag{3.11}$$

where the matrix $\hat{\mathbf{T}}$ is the Markov transition kernel, and the vector $\hat{\mathbf{p}}^{\tau}$ is the probability distribution over system states at timestep τ. The computational cost of each time step is roughly constant under this Markov chain, since each step requires computing the transition rate to all possible next states in Equation 3.4.

The current and fixed point distributions $\hat{\mathbf{p}}$ and $\hat{\pi}$ under this process can be related to the corresponding distributions $\mathbf{p}$ and π under the Markov jump process by scaling by the expected holding time,

$$\pi = \frac{1}{Z}\mathbf{D}\hat{\pi}, \tag{3.12}$$

$$\mathbf{p} = \frac{1}{Z}\mathbf{D}\hat{\mathbf{p}}, \tag{3.13}$$

where $\mathbf{D}$ is a diagonal matrix with the expected holding times for each state on the diagonal, $D_{jj} = \frac{1}{\sum_{i \neq j} \Gamma_{ij}}$, and Z is a normalization constant[8]. Similarly, the evolution of $\mathbf{p}$ relative to these discrete time steps can be expressed by scaling $\hat{\mathbf{p}}$ in Equation 3.10 by the holding time,

$$\mathbf{p}^{\tau+1} = \mathbf{D}\hat{\mathbf{T}}\mathbf{D}^{-1}\mathbf{p}^{\tau}, \tag{3.14}$$

$$\mathbf{p}^{\tau+1} = \mathbf{T}\mathbf{p}^{\tau}, \tag{3.15}$$

where $\mathbf{T} = \mathbf{D}\hat{\mathbf{T}}\mathbf{D}^{-1}$ describes the discrete time evolution of the samples. Since $\mathbf{T}$ and $\hat{\mathbf{T}}$ are related to each other by a similarity transform, they share identical eigenvalues. In order to evaluate the spectral gap, and thus the mixing time, of the Markov jump process in terms of computational time, it is thus sufficient to compute the spectral gap of $\hat{\mathbf{T}}$. We do this for randomly generated toy systems in Section 3.3 and Figure 3.3 and show that MJHMC has superior spectral gap characteristics, indicating more efficient mixing.

Relationship to importance sapling

Using the equivalence between discrete and continuous time processes derived in this section, the process of sampling from a Markov jump process can be seen as a realization of importance sampling. The discrete time process $\hat{\mathbf{T}}$ defines the importance sampling proposal distribution, and the holding times provide the importance weights.

Algorithm

Here, we summarize the Markov Jump HMC algorithm for generating N samples in pseudocode. As with all HMC sampling algorithms, an energy function $E(x)$ (equivalent to $-\log \pi(\mathbf{x})$ plus a constant) and its gradient are required. The three hyperparameters are the leapfrog step size ϵ, the number of leapfrog steps per sampling step M; and the momentum corruption rate β.

Note that computation of $\mathbf{L}^{-1}\boldsymbol{\zeta}$ is only necessary when the last transition made was a momentum flip or randomization. The number of times the gradient is evaluated in an MJHMC sampling step is comparable to that of standard HMC.

Algorithm 1: Markov Jump Process HMC

input : ϵ,M,β,$E(x)$,$\nabla E(x)$, N
output: N samples
$\boldsymbol{\zeta}_0 \leftarrow$ Randomly initialized ;
for $i \leftarrow 1$ **to** N **do**
  Calculate states $\mathbf{L}\boldsymbol{\zeta}_{i-1}, \mathbf{F}\boldsymbol{\zeta}_{i-1}, \mathbf{L}^{-1}\boldsymbol{\zeta}_{i-1}$;
  Compute $E(\boldsymbol{\zeta}_{i-1})$, $E(\mathbf{L}\boldsymbol{\zeta}_{i-1})$, $E(\mathbf{F}\boldsymbol{\zeta}_{i-1})$, $E(\mathbf{L}^{-1}\boldsymbol{\zeta}_{i-1})$;
  Compute transition rates Γ_L, Γ_F, Γ_R using Equation (3.9) ;
  Draw waiting times w_L, w_F, w_R from an exponential distribution, using rates of Γ_L, Γ_F, Γ_R respectively;
  Record holding time for $\boldsymbol{\zeta}_{i-1}$, $h_{i-1} \leftarrow \min(w_L, w_F, w_R)$;
  Set $\boldsymbol{\zeta}_i$ to whichever of $\mathbf{L}\boldsymbol{\zeta}$, $\mathbf{F}\boldsymbol{\zeta}$, $\mathbf{R}\boldsymbol{\zeta}$ had the shortest waiting time ;
end
Resample all $\boldsymbol{\zeta}_i$ using holding times h_i as importance weights ;

3.3 Experimental results

Spectral gap on HMC state ladder

The convergence rate of a Markov process to its steady state is given by its spectral gap[193]. This is the difference in the magnitude of the two largest eigenvalues of the Markov transition operator. We numerically compute this value for randomly generated toy problems in order to compare our mixing rate to that of standard HMC. As all HMC algorithms randomize momentum in nearly the same way, it is expected that their mixing time over a single state ladder is representative of their mixing time over the entire state space. To achieve analytic tractability we restrict our attention to finite state ladders. To avoid edge effects, we attach the top and bottom rungs of the ladder to each other, so that the ladder forms a loop and $\mathbf{L}^k\boldsymbol{\zeta} = \boldsymbol{\zeta}$, where k is the number of distinct rungs. We evaluate the eigenvalues on each state ladder using the similarity relationship to a discrete time Markov chain in Equation 3.14.

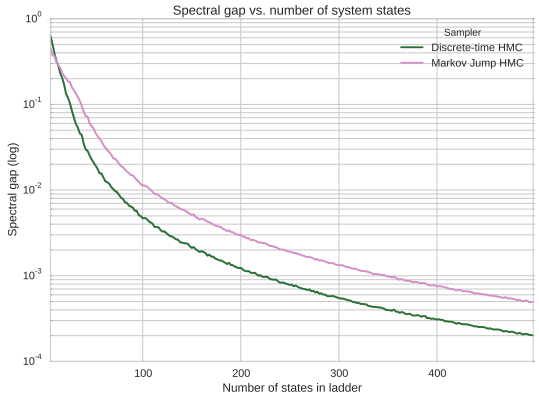


Figure 3.3: Comparison of mixing performance of Markov jump HMC (MJHMC) and standard discrete time HMC. Spectral gap versus size of state ladder. For large state ladder sizes, MJHMC is better by half an order of magnitude.

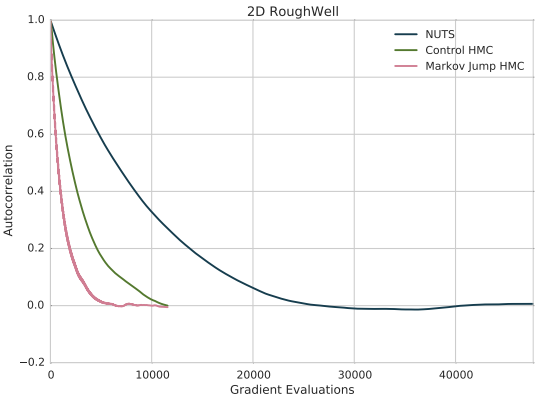


Figure 3.4: Autocorrelation versus number of gradient evaluations for standard HMC and MJHMC for the Rough Well distribution. The hyperparameters found by Spearmint for MJHMC are $\varepsilon = 3.0, \beta = 0.012314, M = 25$ and for control HMC $\varepsilon = 0.591686, \beta = 0.429956, M = 25$.

A comparison of such spectral gaps between Markov Jump HMC and standard HMC is illustrated in Figure 3.3 as a function of state ladder size. We draw the energy for each 'rung' of the state space ladder from a unit norm Gaussian distribution, and average across 250 draws for each ladder size. Figure 3.3 thus shows performance averaged over many randomly generated energy landscapes. MJHMC mixes faster (has a larger spectral gap) for all except the smallest state space sizes.

Autocorrelation on rough well distribution

Explicit computation of mixing time for most problems is computationally intractable. It is common to instead use the rate at which the sample autocorrelation approaches zero as a proxy. As illustrated in Figure 3.4, we compare autocorrelation traces for MJHMC with standard HMC and NUTS on the rough well distribution, and find that MJHMC performs significantly better.

Energy function

The chosen energy function was the 'rough well' distribution from [180]. This distribution provides a good test case because it is as simple as possible, while also presenting both well understood and significant challenges to HMC-style samplers. Its energy function is

$$E(x) = \frac{1}{2\sigma_1^2}(x_1^2 + x_2^2) + cos\left(\frac{\pi x_1}{\sigma_2}\right) + cos\left(\frac{\pi x_2}{\sigma_2}\right), \tag{3.16}$$

where $\sigma_1 = 100$ and $\sigma_2 = 4$. Although this distribution is well conditioned everywhere, the sinusoids cause it to have a 'rough' surface, such that it requires many leapfrog steps to traverse the quadratic well while maintaining a reasonable discretization error.

Hyperparameter selection

As HMC samplers are sensitive to the choice of hyperparameters, we chose optimal settings of the hyperparameters for MJHMC and standard HMC with the Spearmint hyperparameter optimization package [176]. NUTS self-tunes its hyperparameters during burn-in, so we did not perform a hyperparameter search for NUTS.

For each hyperparameter setting, we computed the autocorrelation $C(n)$ as a function of number of gradient evaluations n. We then fit a function $\rho(n; r)$ to this in terms of a complex coefficient $r \in \mathbb{C}$,

$$\hat{r} = \underset{r}{\operatorname{argmin}} \left|\left|\rho(n; r) - C(n)\right|\right|^2 \tag{3.17}$$

$$\rho(n; r) = \operatorname{Re}\left[\exp(rn)\right] \tag{3.18}$$

where the imaginary portion of r corresponds to an oscillatory rate, and the real part corresponds to the decay rate towards 0 autocorrelation. Spearmint is used to find the hyperparameters which make $\operatorname{Re}(r)$ as negative as possible.

We provide additional figures to provide support for our hyperparameter search results in Appendix Figures 3.5, 3.6, and 3.8.

3.4 Discussion

We introduced an algorithm, Markov Jump Hamiltonian Monte Carlo (MJHMC), in which the state transitions in Hamiltonian Monte Carlo sampling occur as Poisson processes in

continuous time, rather than at discrete time steps. We demonstrated that this algorithm led to improved mixing performance, as measured by explicit computation of the spectral gap, by the autocorrelation of the sampler on a simple but challenging distribution.

3.5 Transition rates satisfy balance condition

The continuous time balance condition states that, at the steady state distribution, there is no net change in the probability of states,

$$\left.\frac{\partial p\left(\boldsymbol{\zeta}\right)}{\partial t}\right|_{p(\boldsymbol{\zeta})=\pi(\boldsymbol{\zeta})} = 0. \tag{3.19}$$

In order to demonstrate that we satisfy the balance condition, we evaluate $\left.\frac{\partial p(\boldsymbol{\zeta})}{\partial t}\right|_{p(\boldsymbol{\zeta})=\pi(\boldsymbol{\zeta})}$ using the transition rates from Equation 3.9,

$$\begin{aligned}
\left.\frac{\partial p\left(\boldsymbol{\zeta}\right)}{\partial t}\right|_{p(\boldsymbol{\zeta})=\pi(\boldsymbol{\zeta})} = & -\pi\left(\boldsymbol{\zeta}\right)\left[\frac{\pi\left(\mathbf{L}\boldsymbol{\zeta}\right)}{\pi\left(\boldsymbol{\zeta}\right)}\right]^{\frac{1}{2}} \\
& + \pi\left(\mathbf{L}^{-1}\boldsymbol{\zeta}\right)\left[\frac{\pi\left(\mathbf{L}\mathbf{L}^{-1}\boldsymbol{\zeta}\right)}{\pi\left(\mathbf{L}^{-1}\boldsymbol{\zeta}\right)}\right]^{\frac{1}{2}} \\
& - \pi\left(\boldsymbol{\zeta}\right)\max\left[0, \left[\frac{\pi\left(\mathbf{L}^{-1}\boldsymbol{\zeta}\right)}{\pi\left(\boldsymbol{\zeta}\right)}\right]^{\frac{1}{2}} - \left[\frac{\pi\left(\mathbf{L}\boldsymbol{\zeta}\right)}{\pi\left(\boldsymbol{\zeta}\right)}\right]^{\frac{1}{2}}\right] \\
& + \pi\left(\mathbf{F}\boldsymbol{\zeta}\right)\max\left[0, \left[\frac{\pi\left(\mathbf{L}^{-1}\mathbf{F}\boldsymbol{\zeta}\right)}{\pi\left(\mathbf{F}\boldsymbol{\zeta}\right)}\right]^{\frac{1}{2}} - \left[\frac{\pi\left(\mathbf{L}\mathbf{F}\boldsymbol{\zeta}\right)}{\pi\left(\mathbf{F}\boldsymbol{\zeta}\right)}\right]^{\frac{1}{2}}\right] \\
& - \pi\left(\boldsymbol{\zeta}\right)\beta \\
& + \int d\mathbf{x}' d\mathbf{v}' \pi\left(\mathbf{x}', \mathbf{v}'\right)\delta\left(\mathbf{x}' - \mathbf{x}\right)\pi\left(\mathbf{v}\right)\beta,
\end{aligned} \tag{3.20}$$

where negative terms correspond to probability flow out of state $\boldsymbol{\zeta}$ into other states, and positive terms correspond to probability flow from other states into state $\boldsymbol{\zeta}$. There are only a small number of terms because transitions are only allowed to/from a limited set of states.

We now proceed to simplify and cancel terms,

$$\begin{aligned}
\left.\frac{\partial p(\boldsymbol{\zeta})}{\partial t}\right|_{p(\boldsymbol{\zeta})=\pi(\boldsymbol{\zeta})} =& -\left[\pi(\mathbf{L}\boldsymbol{\zeta})\,\pi(\boldsymbol{\zeta})\right]^{\frac{1}{2}} + \left[\pi(\boldsymbol{\zeta})\,\pi\left(\mathbf{L}^{-1}\boldsymbol{\zeta}\right)\right]^{\frac{1}{2}} \\
& - \pi(\boldsymbol{\zeta}) \max\left[0, \left[\frac{\pi(\mathbf{L}^{-1}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} - \left[\frac{\pi(\mathbf{L}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}}\right] \\
& + \pi(\boldsymbol{\zeta}) \max\left[0, \left[\frac{\pi(\mathbf{L}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} - \left[\frac{\pi(\mathbf{L}^{-1}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}}\right] \\
& - \pi(\boldsymbol{\zeta})\,\beta + \pi(\mathbf{x})\,\pi(\mathbf{v})\,\beta \qquad (3.21) \\
=& -\left[\pi(\mathbf{L}\boldsymbol{\zeta})\,\pi(\boldsymbol{\zeta})\right]^{\frac{1}{2}} + \left[\pi(\boldsymbol{\zeta})\,\pi\left(\mathbf{L}^{-1}\boldsymbol{\zeta}\right)\right]^{\frac{1}{2}} \\
& + \pi(\boldsymbol{\zeta})\left[\frac{\pi(\mathbf{L}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} - \pi(\boldsymbol{\zeta})\left[\frac{\pi(\mathbf{L}^{-1}\boldsymbol{\zeta})}{\pi(\boldsymbol{\zeta})}\right]^{\frac{1}{2}} \qquad (3.22) \\
=& -\left[\pi(\mathbf{L}\boldsymbol{\zeta})\,\pi(\boldsymbol{\zeta})\right]^{\frac{1}{2}} + \left[\pi(\boldsymbol{\zeta})\,\pi\left(\mathbf{L}^{-1}\boldsymbol{\zeta}\right)\right]^{\frac{1}{2}} + \left[\pi(\mathbf{L}\boldsymbol{\zeta})\,\pi(\boldsymbol{\zeta})\right]^{\frac{1}{2}} \\
& - \left[\pi(\boldsymbol{\zeta})\,\pi\left(\mathbf{L}^{-1}\boldsymbol{\zeta}\right)\right]^{\frac{1}{2}} \qquad (3.23) \\
=& 0. \qquad (3.24)
\end{aligned}$$

Therefore the transition rates in MJHMC satisfy the balance condition for $\pi(\boldsymbol{\zeta})$, as claimed.

3.6 Hyperparameter search

Demonstration of optimized hyperparameters

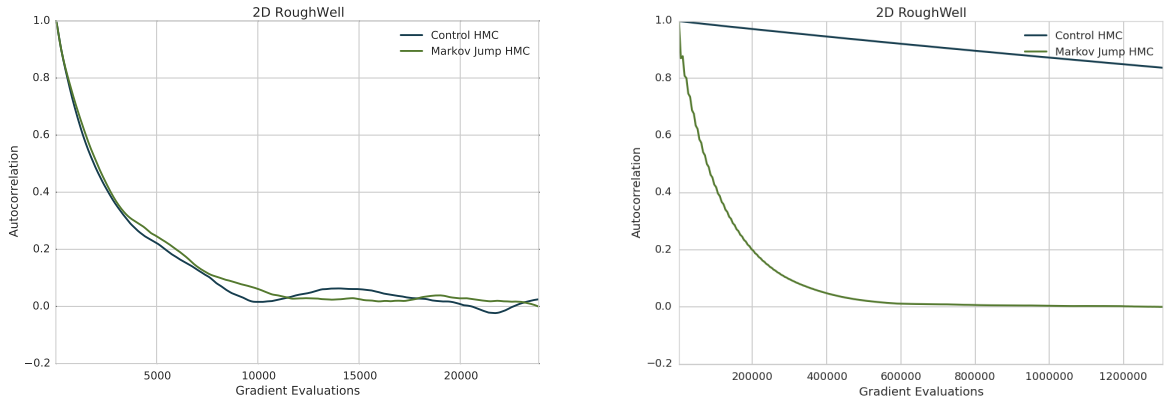


Figure 3.5: Comparison of mixing performance of MJHMC and standard discrete-time HMC with both samplers set to the same hyperparameters *(a)* Both samplers set to $\varepsilon = 0.591686, \beta = 0.429956, M = 25$, the best setting for standard HMC found by Spearmint *(b)* Both samplers set to $\varepsilon = 3.0, \beta = 0.012314, M = 25$, the best settings for MJHMC found by Spearmint

The autocorrelation data illustrated in figure 3.5 demonstrates that Spearmint found effective hyperparameters for MJHMC and standard discrete-time HMC on our chosen energy function. Each sampler outperforms the other when both are set its optimized setting of hyperparameters.

Illustration of Spearmint search

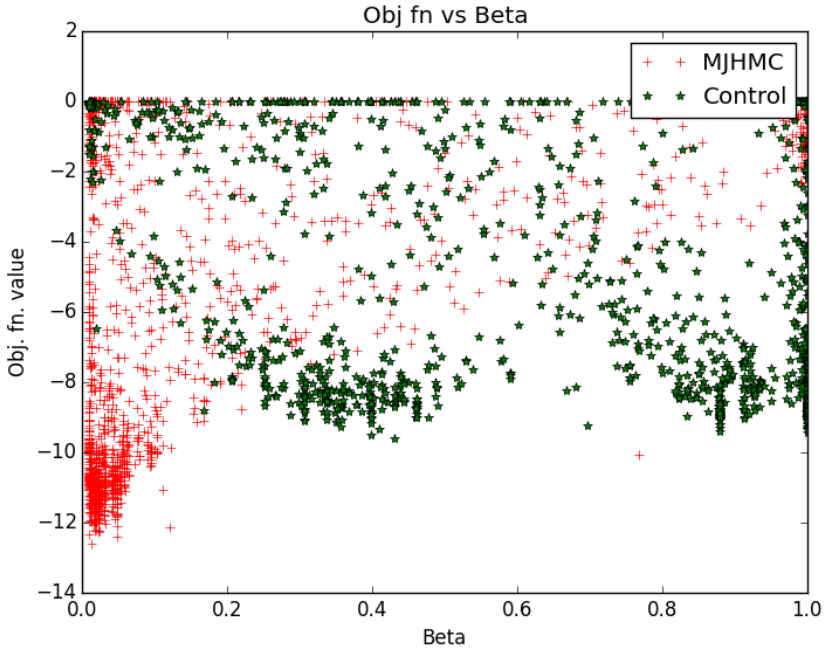


Figure 3.6: Search performance projected onto the β axis.

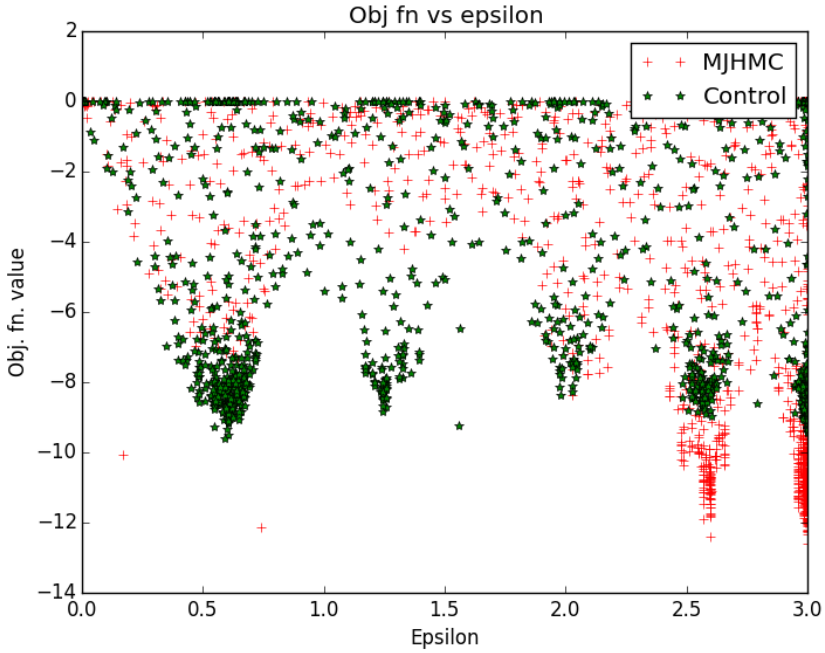


Figure 3.7: Search performance projected onto the ε axis.

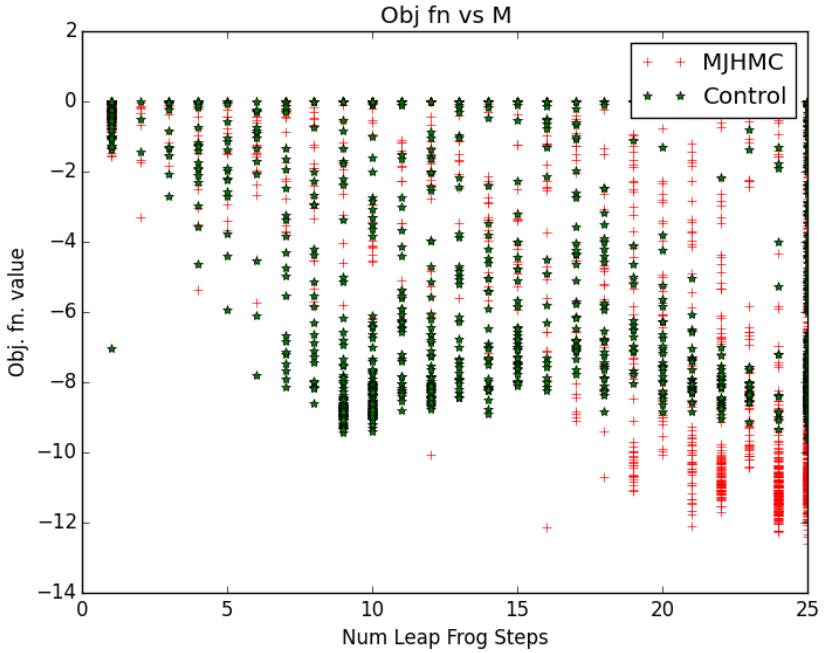


Figure 3.8: Search performance projected onto the M axis.

Figures 3.6, 3.7, 3.8 illustrate the overall structure of Spearmint's search for hyperparameters. The green stars represent a trial setting of MJHMC hyperpameter and the red crosses represent a trial setting of standard HMC hyperparameters. The y-axis represents the value of the objective function for each trial setting. It can be seen that in 3.6 that MJHMC chooses smaller β values which suggests wanting to corrupt momentum more slowly as compared to the control case. It can also be seen from Figures 3.7, 3.8 that MJHMC prefers larger steplengths for the integrator (ϵ) and steps (M) .

3.7 Derivation of equation 3.11

First we calculate $P(\tau_2 \leq \tau_1)$ where τ_i is drawn from $\text{Exp}(\lambda_i)$ for $i = 1, 2$:

$$\begin{aligned}
P(\tau_2 \leq \tau_1) &= \int_0^\infty P(\tau_1 = tdt)P(\tau_2 \leq t)dt \\
&= \int_0^\infty (\lambda_1 \exp(-\lambda_1 t)) \int_0^t (\lambda_2 \exp(-\lambda_2 \tau))d\tau dt \\
&= \lambda_1 \int_0^\infty \exp(-\lambda_1 t)\left[1 - \exp(-\lambda_2 t)\right] dt \\
&= \lambda_1 \int_0^\infty \exp(-\lambda_1 t) - \exp(-(\lambda_1 + \lambda_2)t)dt \\
&= \lambda_1 \left[-\frac{1}{\lambda_1}\exp(-\lambda_1 t)\Big|_0^\infty\right] - \left[-\frac{1}{\lambda_1 + \lambda_2}\exp(-(\lambda_1 + \lambda_1)t)\Big|_0^\infty\right] \\
&= \lambda_1 \left[\frac{1}{\lambda_1} - \frac{1}{\lambda_1 + \lambda_2}\right] \\
&= 1 - \frac{\lambda_1}{\lambda_1 + \lambda_2} \\
&= \frac{\lambda_2}{\lambda_1 + \lambda_2}
\end{aligned}$$

Let $\tau_{i,j}$ be drawn from $\text{Exp}(\Gamma_{ij})$. Then

$$\begin{aligned}
P(\boldsymbol{\zeta}_j \mid \boldsymbol{\zeta}_i) &= P(\tau_{i,j} = \min\{\tau_{i,1}, \tau_{i,2}, \ldots \tau_{i,n}\}) \\
&= \prod_{k \neq j}^{n} P(\tau_{i,j} \leq \tau_{i,k}) \\
&= \prod_{k \neq j}^{n} \frac{\Gamma_{i,j}}{\Gamma_{i,j} + \Gamma_{i,k}}
\end{aligned}$$

3.8 Acknowledgements

The work in this chapter was performed in collaboration with Andrew Berger , and Jascha Sohl-Dickstein with advise from Mike Deweese. This work was presented at the MCQMC conference held at Stanford University in Aug 2016

Chapter 4

Hamiltonian Monte-Carlo-inspired Sampling in analog devices

Statistical inference is an important and challenging problem for both natural and man-made systems. This inference often necessitates the approximation of probability distributions. Markov Chain Monte Carlo (MCMC) methods are powerful tools for approximating, via sampling, complex distributions of interest. These methods are often slow on digital computers, but sampling in analog systems can be significantly faster. We present a framework – inspired by a particular type of MCMC method, Hamiltonian Monte Carlo (HMC) – that uses the natural dynamics of charge passing through a circuit to sample from analog devices. We provide a simple example circuit along with some resulting distributions and discuss key open challenges with this framework.

4.1 Introduction

Probabilistic models are useful for describing random variables of interest, such as parameters characterizing the environment of an agent, biological or otherwise. In a Bayesian framework, model parameters by θ, the probability distribution over the hypothesis (model parameters) $\mathcal{H}$ given the input (data) x is denoted by $P(\mathcal{H}|x) = \frac{P(x|\mathcal{H})P(\mathcal{H})}{P(x)}$. This is also known as the posterior probability distribution.

This is often solved by collapsing the distribution to a point estimate using the maximum a posteriori estimate (MAP). For many real world scenarios, however, choosing the MAP estimate results in a poor optimal solution as this prevents downstream systems from leveraging the power of multiple hypotheses. To maintain the full state of the system, one can generate samples to approximate the entire distribution.

In practice, sampling is often done using Markov Chain Monte Carlo (MCMC) methods. There is a large literature on sampling in digital systems, with two of the most common methods being Gibbs [174] and Metropolis Hastings [80]. One fast sampling method for high dimensional spaces is Hamiltonian Monte Carlo (HMC) [138]. However, on digital systems,

this is typically quite slow [127]. Thus, inspired by HMC, we aim to leverage the natural, Hamiltonian dynamics of a circuit to sample in analog systems. To the best of our knowledge, this work is the first to show this connection.

4.2 Sampling in analog devices

The example we consider here is a series LRC circuit. Specifically, we can drive this LRC circuit with a simple δ-pulse train and sample the value of charge on its capacitor. Following Kirchhoff's laws, the time-dependent charge on the capacitor in response to a singular δ-pulse is given by:

$$q\left(t\right) = Q_0 e^{-\frac{t}{\tau}} \cos\left(\omega t\right), \tag{4.1}$$

where $\tau = \frac{2L}{R}$ and $\omega = \sqrt{\omega_0^2 - \frac{1}{\tau^2}}$. The dynamics of charge for this capacitor satisfy the two requirements for a valid sampler [180]: (i) *leaving the interest distribution invariant* and (ii) being *ergodic*. A histogram of these charge samples will form an (un-normalized) probability distribution. This distribution corresponds to the number of solutions to the charge dynamics equation at every value in its range. By modifying the circuit layout, components, and driving input, the circuit dynamics and corresponding histogram can be fit to the distribution of interest. We currently do not have a closed form expression for the charge distribution as a functional of the charge dynamics, but we are actively pursuing this line of work.

Figure 4.1a illustrates the dynamics resulting from one choice of circuit element values and pulse separation. Figure 4.1b shows the corresponding distribution of capacitor charge samples over this duration. By modifying the time between pulses (Figure 4.1c), we can easily generate different types of distributions (Figure 4.1d). One simple way to extend this model to high dimensional spaces is to have multiple LRC circuits generating independent samples for each dimension of interest. For example, by combining the dynamics in Figure 4.1a and Figure 4.1c for two different dimensions we can generate a 2D bi-modal distribution (Figure 4.1e). Importantly, this is a challenging distribution for many sampling methods, having a 'thin bridge' of probability mass [180]. This parallelization only works in cases where the dimensions of interest are independent of each other. If the variables are coupled, more complicated dynamics would need to be explored. To ensure that all energy contours are sampled from randomly (and not deterministically), we uniformly sample a number using an RC circuit. This number is used as a delay to sample from the capacitor for each dimension. This is equivalent to randomizing the momentum in an HMC [138].

We then consider the problem of learning the circuit parameters for a given distribution. To do this, we present a method presented below.

Sampling is important for many real world applications, and it may even allow the brain to exploit the dynamics of neurons to represent probability distributions [151]. Here, we present a novel way to sample using the natural dynamics of analog circuits. We illustrate our approach by sampling from a few distributions using a simple LRC circuit, and we believe

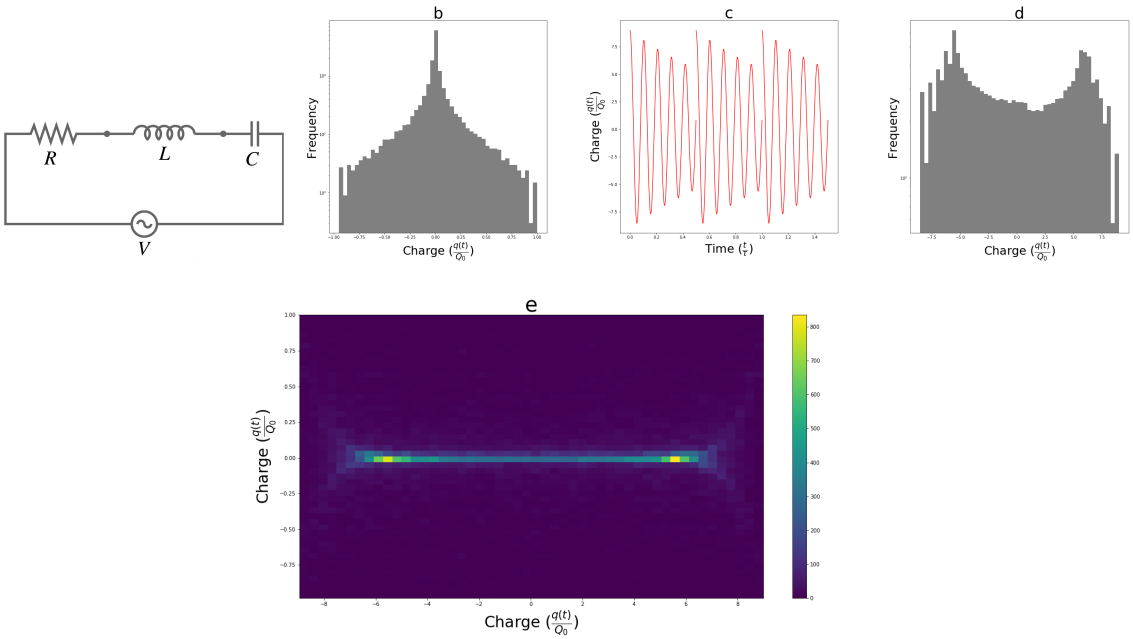


Figure 4.1: (top left a): Charge dynamics in an LRC circuit. The horizontal axis represents dimensionless time; the vertical axis represents the instantaneous value of the dimensionless charge on the capacitor. (top middle b): Histogram representing the distribution of charge for the dynamics specified in 1a. (top middle c): Similar to 1a, this plot describes dynamics with a different time between δ-pulses. (top right d): The corresponding histogram for the dynamics specified in 1c. (bottom e): A 2D density plot of samples generated by simultaneously using the dynamics specified in 1a and 2a for two different dimensions.

Algorithm 2: Fitting circuit parameters for a given distribution

Given circuit parameters:$\psi = \{L, R, C, \dots\}$;
Given a function that generates distribution: $P := f_{circuit}(\psi)$;
Let P_θ be the distribution generated by a circuit ;
Further let $\theta_T := \theta(\psi_T)$, $\theta_M := \theta(\psi_M)$ be *target* and *model* samples ;
Init: $error \leftarrow 1e + 20$;
Result: Return: θ^*_M
while $error < \epsilon$ **do**
    The kernel density estimate is computed by$P_\theta = \sum_n K\left(\theta^{(n)}; \sigma\right)$;
    $K\left(\theta^{(n)}; \sigma\right) = \frac{1}{\sqrt{2\pi}\theta^{(n)}} e^{-\frac{(x-n)^2}{2\sigma^2}}$;
    $\theta^*_M \leftarrow \min_{\theta_M} D_{KL}(P_{\theta_M} || P_{\theta_T})$;
    Compute *error* from the updated parameters ;
end

that this is a promising approach to explore inference in analog devices. Importantly, this framework is not limited to electronic circuits and could be applied to studying the dynamics of neural circuits.

4.3 Discussion

There are many future directions one can explore with this framework. One direction is to study sampling dynamics for a complex analog system, like an analog Hopfield network. Another direction is to explore sampling from a general class of distributions (i.e. from specific families) by having a model that could iteratively learn the circuit parameters required to generate said distributions (e.g. the inductances, capacitances, and resistances). Lastly, we believe this approach can be extended to naturally utilize (and not simply work around) the stochasticity inherent in natural systems – whether they be, say, the dynamics of spiking neurons or of novel electronic substrates – thus providing a more efficient framework for computation.

Indeed, [190] have shown that coupled oscillators can actually be used to implement *Ising* machines using continuous dynamics in analog devices but using *injection locking* to force the system into binary states. This leads us to believe that this direction of work is very promising to explore analog devices and their dynamics for computation.

We have only explored certain simple circuit elements and it would be interesting to consider more complex, non-linear circuit elements such as memristors [34] to build these circuits.

4.4 Acknowledgements

The work in this chapter was performed in collaboration with Ryan Zarcone and Mike Fang with advise from Bruno Olshausen and Mike Deweese. Original conversations about this idea were also had with Jascha Sohl-Dickstein and Jim Crutchfield. This work was presented at the first Cognitive Computing conference at Hannover, December 2018

Chapter 5

Investigating deep reinforcement learning for grasping objects with an anthropomorphic hand

5.1 Introduction

Grasping is among the basic building blocks of object manipulation. It is a well established and important problem in robotics that has garnered a lot of research interest [137, 44, 68, 110, 158, 81, 99, 152, 119, 73, 126]. While past works have made excellent progress in demonstrating the ability to grasp even from high-dimensional observations such as images, they have employed rather simplistic two finger parallel jaw grippers. Pragmatically, it can be argued that if the end goal is simply to grasp an object of interest, parallel jaw grippers are sufficient for grasping most objects. However just grasping an object is not a very useful skill by itself and humans often perform complex in-hand manipulation of grasped objects for achieving desired goals. The ability to perform human like in-hand manipulation has generated a lot of interest in design and control of anthropomorphic hands such as the Shadow Hand, the ADROIT suite [107], the UW Hand [194], RBO 2 hand [42] and many others [160, 163].

The superiority of anthropomorphic hands to parallel jaw grippers for in-hand manipulation comes at the cost of difficulty in finding a policy to control them in a desirable fashion. Past work on controlling anthropomorphic hands has relied on imitation [160], human demonstration [74, 108] or pre-defined grasp types [160] as priors for inferring control strategies or has exploited compliance and under-actuation [142, 43] for reducing the number of degrees of actuation which in turn simplifies the control problem. However compliance inevitably leads to loss in precise knowledge of the object state. To account for this more recently the use of touch and other sensory measurements have been under investigation for recovering state information [189]. As the observation space becomes richer and the complexity of tasks increases, constructing analytical models for control is going to become progressively harder.

Figure 5.1: The left column shows a translucent visualization of the anthropomorphic hand based on the SHAP procedure [123] used in our experiments. The palm is facing away from the table and location of joints is shown with blue arrows and touch sensors are shown in green. The hand has 22 joints out of which 13 are actuated. Several joints are tendon coupled. In addition there are three degrees of freedom for translation along x, y and z axis. Overall these sixteen degrees of actuation allow for complex hand movements. The right two columns show three objects of different geometrical and material properties placed in arbitrary position and pose with respect to the hand. We show that is possible to learn a policy for grasping these objects using model free reinforcement learning.

While using human demonstrations to learn control policies would always be an alternative, the approach is not satisfactory and for many interesting tasks collecting demonstrations is known to be a hard and tedious process.

Reinforcement learning provides a general paradigm for policy learning without requiring any models of the actuator or the environment. However, most current reinforcement learning algorithms [124, 168, 135, 134] rely on random walk in the action space for exploration which makes it non-trivial to control actuators with large number of degrees of freedom. It is therefore not surprising and to the best of our knowledge, learning robust grasping behavior from scratch using model free reinforcement learning with a joint controlled anthropomorphic hand in the face of changing environmental conditions and sensory noise has not yet been demonstrated. Given that it is extremely challenging to manipulate objects with an anthropomorphic hand, in this work we start with learning the grasping primitive and investigate the limitations and successes of current reinforcement learning techniques towards this end.

We use a simulated version of a prosthetic hand developed for SouthHampton Hand

Assessment Procedure (SHAP [123]) (as shown in Figure 5.1) to learn a policy using 16 degree of freedom actuation space for grasping objects from proprioceptive and object state observation. Because the mapping between the observations and control signals is complex we use deep neural networks to represent the policy. We find that when appropriately tuned, a state of art policy learning method known as trust policy region optimization (TRPO [168]) is able to learn policies for controlling the SHAP arm to successfully grasp objects from a narrow range of positions. We analyze several properties of the learned policy and find that they are able to grasp objects with novel geometries, generalize to positions that were outside the training range and are robust to sensory noise and changes in material properties of the object such as mass. We also investigate whether including of tactile observations lead to learning of more robust grasping strategies or not. Overall our results show that it is possible to control complex anthropomorphic hands with deep reinforcement learning. We hope that our work will serve as a strong baseline for future methods and open doors for trying more complex tasks using dexterous manipulators.

5.2 Experimental Setup

We use a simulated model of the anthropomorphic hand used as part of SouthHampton Hand Assesment Procedure (SHAP) test suite [123] (see Figure 5.1). The SHAP procedure was established for evaluating prosthetic hands and arms. With this idea in mind, prior work [153] built a prosthetic arm which could theoretically perform all useful human hand movements. Based on this hand and the DARPA Haptix challenge[192], a simulated model of a similar hand (but with fewer sensors) [106] was built using the Mujoco physics engine [184]. This model was made publically available and we use this for all our experiments.

The hand has five fingers and 22 joints out of which many are tendon coupled, i.e. actuating one joint actuates a set of other joints. For example, curling the tip of a finger automatically actuates the other two joints on the finger so that the finger moves towards the palm. Because of these couplings, the resultant dynamics can be quite complex and articulated. Out of the 22 joints, thirteen are actuated. Out of these thirteen, ten joints control the motion of fingers and the other three control the rotation of the hand. Additionally, there are three degrees of motion along the (x, y, z) axis and therefore overall 16 degrees of actuation. The hand is controlled by setting the position of these 16 actuators.

We use the SHAP arm to learn policies for grasping a single object kept on a table. We use three objects for training, namely a cube, a sphere and an ellipsoid shown in Figure 5.1. We experiment with two different settings - (a) training individual policies for grasping different objects and (b) training a single joint policy for grasping the three objects in the training set. We then test the robustness of these policies to sensory noise, changes in material properties such as mass of objects. Finally, we test how well the learned policies generalize to four novel objects - cylinder, coin, a hollowed cylinder (can) and a screw driver. For the rest of this document, when we refer to experiments we refer to simulated experiments in Mujoco [184].

Figure 5.2: Grasp trajectory of a policy learnt to grasp spheres in various initial positions with respect to the hand. The first two rows show cases where the policy suceeds Left to right, we see the trajectory unroll in six sub images. The general learnt policy is to go under the object and scoop the sphere into the palm. In some cases, it just balances the object in other cases it wedges the object between fingers and sometimes the thumb. The last two rows show cases where the policy fails.

5.3 Model

Preliminaries

Reinforcement learning problems are specified by - a set of states $\mathcal{S}$ which determines the state of the agent, the set of actions $\mathcal{A}$, a transition function $\mathcal{T} \rightarrow \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ that takes the current state and action to provide the next state, a reward function that produces a scalar value $\mathcal{R} \rightarrow \mathcal{S} \times \mathcal{A}$ for a given state, action pair and a discount factor γ. When the transition function $\mathcal{T}$ is explicitly specified it is known as model based control and when it is implicitly specified through the world or simulator then this problem is often referred to *model-free control*. In this work we employ model-free learning because of its generality. Learning model

free policies in continuous actions and state space is a very challenging problem, especially as the dimensionality of the states and actions grows.

Formally, the problem of reinforcement learning can be stated as the problem of learning a policy π_θ (i.e. a policy with parameters θ) for choosing actions given the current observation maximizes the expected reward:

$$\max_{\pi_\theta} \mathbb{E}_{s_0,a_0\ldots} \sum_{t=0}^{\infty} \gamma^t r(s_t) \tag{5.1}$$

There are two broad class of model free reinforcement learning algorithms - (a) policy gradient based methods that directly try to maximize the expected reward and (b) Q-learning based methods that maximize the Q-function for determining a policy. In complex problems such as ours, the mapping between observations and actions is highly-non linear and therefore high capacity learners such as deep neural networks are often employed for modelling the policy. In this work, we investigate two policy learning methods: (a) Trust region policy optimization (TRPO) which is a policy gradient method that has produced impressive results on ATARI games and simulated humanoid locomotion [168] and (b) Deep deterministic policy gradients (DDPG [124]) that is a variant of Q-learning. In our experiments we found that DDPG was not robust and we were unable to learn grasping behaviors using DDPG. Because DDPG was not successful, we do not describe it in detail, but instead focus on TRPO.

The main idea behind TRPO is to use the current policy π_θ to collect data from multiple trajectory roll outs. These trajectories are used to compute the update to the parameters θ that maximize the expected reward. Since even small changes in θ can lead to large changes in the policy when using powerful function approximators such as deep neural networks, the policies are only allowed to change by a small amount (i.e. within the trust region) by constraining the the KL divergence between the updated and old policy to be small. The number of trajectories used to compute the policy update (i.e. the policy gradient) is referred to as the *batch size* parameter of TRPO. A larger batch leads to a better estimate of the policy gradient. TRPO learns a stochastic policy which is often represented with a gaussian distribution. The initial standard deviation of the policy governs initial exploration and is a key parameter that needs to be tuned for achieving good performance.

In addition to the parameters mentioned above, deep reinforcement learning methods are also known to be quite sensitive to the random seed used to initialize the weights of deep neural network weights (i.e. parameters). One reason is that initial weights influence exploration, which in turn effects the overall performance of the system. For the sake of repeatability, in all our experiments we report the mean reward across three runs of the RL agent with three different seeds.

Observation Space

The observation space consists of the internal states of the hand and the state of the object. As described in [106], the hand state is comprised of joint position and velocity sensors on all

22 joints; force, position and velocity sensors on the 13 actuators; touch sensors as shown in Figure 5.1 and inertial measurement units on all 5 finger tips. We augment this feature with the location of the object centroid in the global coordinate frame. Overall the observation space had 53 dimensions. In some experiments we made use of touch sensors on the hands and in this case our feature space had 72 dimensions (i.e. 19 touch sensors).

The observation described above was fed as input to the neural network representing the policy. We normalized the observation by computing the mean and standard deviation of the observations collected using 1000 rollouts of a randomly initialized policy.

Reward Structure

In order to grasp an object the hand must reach the object and then lift it. One way to specify the reward for grasping is the height of the object from the table ($\mathcal{D}_{ot}$). However, this reward is sparse because random exploration is unlikely to result in object grasps by chance. Unsurprisingly, with this sparse reward we were unable to learning grasping policies. In order to encourage the hand to grasp the object we shaped the reward to include negative of the distance from the palm of the hand to the object (i.e. $-\mathcal{D}_{op}$).

Simply defining the reward as $\mathcal{D}_{ot} - \mathcal{D}_{op}$ is not appropriate because initially $\mathcal{D}_{ot}$ will be zero and the reward will be maximized by moving close to the object. In the process of minimizing the distance to the object, the policy might land up in a local minima where it is encouraged to reach as close as possible to the object at the cost of limiting exploration which might be required to learn grasping behavior. This is in fact what happens and we find that with such a reward function, the hand learns to get under the object and then flicks the object with the maximal torque to raise it away from the table. While this results in positive reward signal, the hand fails to grasp the object.

We mitigate this issue by encouraging the hand to go close enough to the object so that it is able to explore interesting object interaction behavior without getting stuck in a local minima. We achieve this by defining a threshold on the distance (ϵ) so that when the hand is within a distance of ϵ from the object it is provided with no negative penalty. Our reward function, therefore looks like:

$$\begin{aligned} r(t) &= -0.1 \times \mathcal{D}_{op} \quad \mathcal{D}_{op} \geq \epsilon \\ r(t) &= \mathcal{D}_{ot} \quad \mathcal{D}_{op} < \epsilon \end{aligned} \tag{5.2}$$

We set ϵ to be 0.15, which was about approximately two times the radius of the sphere object used in our experiments. We found that with this reward function it was possible to learn good grasping policies. In our experiments, each episode was of length 500 time steps.

5.4 Results

We used trust region policy optimization (TRPO) [168] as the policy learning algorithm. We found that it was hard to train the off-policy methods such of deep deterministic policy

batch size	relative space
40k	169.499 +/- 96.98
60k	332.57 +/- 21.04
80k	281.887 +/- 36.65
120k	349.90 +/- 38.60

Table 5.1: Table comparing the mean (across three seeds) average reward per episode during training for different batch sizes. The standard deviation is provided after the +/- symbol.

gradient (DDPG) [124] for this task. For TRPO, the neural network policy was represented by a Gaussian multi-layer perceptron (MLP) with two hidden layers with sizes of 32 units and 32 units. We experimented with varying the size of the network by testing networks of size 64 and 16 but found that they performed comparably (64 units) or worse (16 units). Based on these preliminary results we used two layer neural networks of size (32, 32) for the rest of our experiments.

We found that two parameters were critical to training: (a) batch size (b) the initial standard deviation of the Gaussian policy and (c) normalizing the observations. We varied the initial standard deviation of the Gaussian across multiple experiments and settled on a value of 3.0. Any lower value of the standard deviation led to policies ending up in initial local minima that were hard to escape from later on. We fixed the episode length to be 500 time steps long. We empirically found that this was a long enough trajectory to reach to almost any part of the table with the torques that were being generated by the model. To normalize the observations, we unrolled 1000 random episodes and computed their means and standard deviations. These were then used to normalize the observations during learning. We tried experiments without normalizing the observations and the model failed to learn. Given the seed sensitivity of many Deep-RL algorithms, we were cautious and ran multiple (at least three seeds) for each experiment.

Figure 5.2 shows successful and unsuccessful grasps with our model. It is particularly interesting to observe the failure modes of the model. It finds it difficult to grasp objects that have been placed to the right and behind the hand, since this requires the hand to incur a penalty to orient itself before it can grasp successfully. It also fails to grasp if it cannot get underneath the sphere. It would be interesting to explore couplings in the hand such that it would enable more dexterous movement so that it could learn to do fine motor control.

Batch size

To test the robustness of the learning algorithm and the feature space to batch sizes, we swept the batch size parameter. We tested the following batch sizes - 40k, 60k, 80k and 120k.The performance scaled linearly with the number of trajectories TRPO saw. The results of this experiment is summarized in Table 5.1

Figure 5.3: Figure showing some of the objects and their grasps by the model (snapshot at different time intervals). Left to right, a can, screw driver, cylinder, coin, ellipsoid and cuboid are shown. The table is bounded by four walls (green in color) that can be seen

Multiple Objects and Generalization

Given the relative success of grasping a single object, we then proceeded to test the hypothesis of grasping multiple objects. Here, we wanted to see how our setup fared at grasping multiple objects. In addition, we hoped to verify if grasping multiple objects actually made it faster or generalized better to grasping novel objects.

We then setup an experiment using the relative feature space to grasp multiple objects - a sphere, cuboid and an ellipsoid at different locations on a table. We set the batch size for this experiment to be 80,000 samples. We show that we are able to learn a policy to grasp for multiple objects as well. Some visualizations from this experiment can be seen in Figure 5.3. We see that it learns to grasp multiple objects it had not seen during the training. Some of them share geometric properties.

We hypothesized that there was some generalization that was possible to other objects given a successful grasping policy with only a single object. We show the results of this in Table 5.2. In column one, we see the results of a policy learnt on a sphere with batch size set to 80,000. The rows describe the success of the model on approximately 125 samples of seven different object categories. We see that the symmetry of a sphere lends itself to grasping a cuboid easily and also to some extent grasping a coin.

To test the generalization of the model to novel objects not used in training we trained two models. One where, we gave the model only a sphere as the object to be grasped. In the second case, we trained the model on three objects - a sphere, ellipsoid and a cuboid. We gave the learnt model novel objects such as a screw driver, a can, an ellipsoid and a cylinder. We show that the model trained on multiple objects generalizes to novel objects better. We report the success of the experiment in Table 5.2. We note that learning to grasp multiple objects is possible with our model and training scheme. Though in both cases it struggles to grasp a can which possibly requires a very different grasp. Generalization to multiple novel samples is possible to some extent when objects seem to share similar underlying geometric properties.

Training with density variations

In the real world similar looking objects rarely have the same mass. So if we had a policy that learns really well only when the physics of the object or world is heavily constrained

Object Type	Sphere Success Rate	Multi-Obj Success Rate
cuboid	0.92	1.0
sphere	0.85	0.99
ellipsoid	0.01	0.98
cylinder	0.17	1.0
can	0.0	0.06
coin	0.46	0.47
screwdriver	0.29	0.97

Table 5.2: Table describing the success of grasping an object and lifting it off the table. In the sphere case, the model was trained only on the sphere as an input. In the multi-obj case the first three rows are objects that the model were trained on. The remaining four are novel objects not present in the training set for both conditions. Success rate is defined here as the number of times the object was at a distance of 0.7 or greater across all episodes for that object. In our simulation this usually meant the object must have been grasped reasonably for the object to be that far from the table. We simulated 1000 episodes and this led approximately to 125 episodes per category

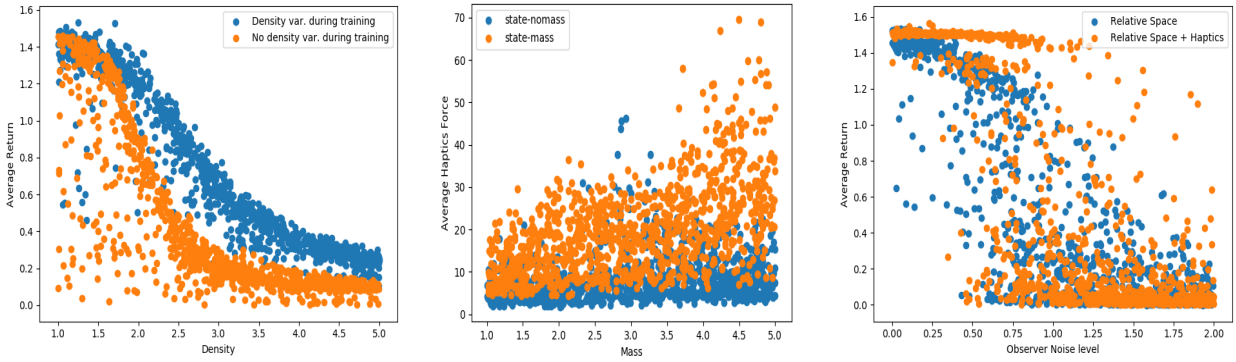


Figure 5.4: **Left:** A scatter plot comparing the average reward (y-axis) and the density (x-axis) for models trained on density variation (blue) versus one that is not trained on density variation (orange). We ran a 1000 samples for each condition to populate the graph. **Middle:** A scatter plot comparing the average force (y-axis) and the density (x-axis) for models trained on density variation (blue) versus one that is not trained on density variation (orange). We ran a 1000 samples for each condition to populate the graph. **Right:** Scatter plot showing the relationship between varying observer noise level and average return. Plotted on the x-axis is the observer noise sampled from a uniform distribution with range 0 to 2 on the x-axis. On the y-axis is the average return across an episode. Two feature spaces were compared - with (orange) and without (blue) haptics as feature spaces. We ran a 1000 samples for each condition to populate the graph

then it might not generalize very well to more complicated problems of grasping. To verify

this we compare how models learn to grasp when they are provided with training data with variations in the density. We take two models of grasping a sphere with random initialization. In one model we varied the mass (density) during training and in the other, we did not. The results of this experiment can be seen in Figure 5.4 (Left). The total force exerted between the two conditions are also different as seen in Figure 5.4 (Middle). There is a noticeable difference in performance when we train under certain kind of perturbations. If our larger goal is to build models of grasping that are robust to variations in geometric shape and noise levels, testing these sort of hypothesis could be valuable in our understanding to training robust grasping policies.

Sensory Noise in training

The previous experiment of varying mass led us to explore adding noise to our observations itself. We tested the generalization of grasping when we add noise to the observations (more than what it was trained with). In addition, we explored the role of haptics towards robust grasping by training a model with haptics features as well. The SHAP MPL arm that we use in our project is equipped with 19 pressure sensors described in green in Figure 5.1. We took the following two models with two different observation spaces - (a) relative distance space as mentioned previously (b) relative distance along with haptic feature space. We trained these models with a noise standard deviation $\sigma = 0.1$. During test time, we sampled noise from a uniform distribution $\eta \in (0, 2)$ and plotted this noise against the Average Return. The results can be seen in Figure 5.4 (Right).

From Figure 5.4 (Right) we see that the haptics feature space is slightly more robust to changes in noise. There are many ways in which this result could be made stronger. For example, the density and type of haptics sensors could greatly effect the kind of features a hand may extract from the environment.

5.5 Conclusion

In this work we show that it is possible to learn robust grasping policies from anthropomorphic hands using model free reinforcement learning algorithm known as trust region policy optimization (TRPO). We found that it was critical to tune the batch size and exploration parameters for achieving good performance.

Furthermore we found that the learned policies were not object specific and when trained with three objects, our system was able to grasp four novel objects with quite different geometries as compared to the training set of objects. Further investigation revealed that the learned policies were robust to noise in sensory observations and variation in object properties such as mass. Finally we found that policies trained with touch sensing were robust to sensory noise than policies trained without touch sensing.

5.6 Acknowledgements

The work in this chapter was performed in collaboration with Pulkit Agrawal with advise from Sergey Levine, Jitendra Malike and Mike Deweese. It was presented at the 2017 Neural Information Processing Workshop in the Deep Reinforcement Learning symposium

Chapter 6

Cross modal object identification using vision, tactile sensing, and curiosity

Combining information from partial observations across multiple sensory modalities to execute goal directed actions is a key aspect of human intelligence. Further, we know human agents are very easily able to translate the task communicated in one sensory domain (say vision) into a representation that enables them to complete this task when they can only sense their environment using a separate sensory modality (say touch). Towards this end, we consider the case of intrinsic curiosity for exploration in high dimensional continuous spaces.

Consequently, in order to build agents with similar capabilities, in this work we consider the problem of a retrieving a target object from a drawer. The agent is provided with an image of a previously unseen object and it learns to explore objects in the drawer using only tactile sensing to retrieve the object that was shown in the image without receiving any visual feedback.

Success at this task requires close integration of visual and tactile sensing along with learning to explore the object. We present a method for performing this task in a simulated environment using an anthropomorphic hand. We hope that future research in the direction of combining sensory signals for acting will find the object retrieval from a drawer to be a useful benchmark problem.

An important aspect of human intelligence is the ability to perform robust, repeatable actions with only partial observability. So, an important question for robust behavior is determining what an agent should observe and how it should go about doing this? For example, if an object (keys, silverware) were to be retrieved from a cluttered drawer then we would have very little difficulty as we match an internal model of the object with that being generated by active tacticle exploration. Similarly, exploring a different pose of a coffee mug from a shelf to pick up your favorite mug. *The key ingredient in these tasks is active exploration.* Our primary contributions in this work can be stated as – **employing**

curiosity based exploration in high dimensional continuous spaces for learning to discriminating objects using tactile sensing

Active Learning

The field of active learning addresses the specific problem of learning to acquire samples for a specific goal, reward or task. For example [93] present a way to categorize objects by learning policies that optimally help in recognition. In contrast, [3] phrase the problem as one of trying to learn to poke. [52] develop a method for robots to predict changes to the environment due to its own actions. The field of active learning is indeed old dating back to the 1800s [18]. It found its way into modern day computer vision with work by [5] where they show that an active observer can solve basic vision problems better than passive ones. [16] present an eigenspace based approach to reduce the number of views required for recognition.

Curiosity and RL

An important aspect of Reinforcement Learning is an agent's ability to effectively explore its state space. This is typically modeled as the exploration aspect of the policy. These can be modeled in numerous ways such as heuristics, stochasticity, etc all while trying to maximize either rewards or other measures. There are many obvious limitations and challenges of using explicit extrinsic reward functions for exploration such as appropriate shaping of rewards, state space of the actions and observations. Intrinsic curiosity is subfield of RL that deals with coming up with a policy that tries to maximize a measure that is intrinsic to the agent i.e. depends only on the agent's existing internal representations.

We refer the readers to [148, 147] for a good summary of existing work in this field. [165] uses the idea of surprise and compression for intrinsic curiosity. More recently, [150] proposed the idea of Intrinsic Curiosity Module (ICM). In their work, they employ a latent embedding space that is used to predict the next state (forward model) of the world. A high error suggests that the agent is visiting a state that it has not previously seen, future visitations to such states should reduce the prediction loss as the forward model improves.

Multimodality

Different sensory modalities provide a different view of the same underlying reality. The ability to transform between sensory modalities therefore provides an interesting way to learn useful representations of sensory inputs. Recent work in self-supervised learning has made extensive use of this observation and shown that useful visual features can be learned by predicting, from images, corresponding sounds [149], ego-motion [3, 94], depth or even predicting color values from grayscale images [201].

In addition to learning feature representations, another and possibly more critical use of sensing from multiple modalities is performing goal directed actions in partially observable settings. In the running example of retrieving objects from a drawer, the agent receives only the image of the object as input and in absence of any light source in the drawer, the agent solely relies on its tactile sensing to find the object. Other examples are a pedestrian getting alerted when she hears the sound of a car coming from the back or animals in the jungle being alerted of a tiger behind the bushes by the sound of the movement. Yet another example showing close integration of two modalities (vision and touch) is a study that found

it became almost impossible for human participants to perform the seemingly trivial task of picking up a matchstick and lighting it when their hands were anesthetized [96].

Haptics

One of the earliest works presenting haptics as a sensory modality to explore the world was by Gibson ([60]).Gibson showed that object recognition dramatically decreased when one could not actively interact with an object.

Lederman and colleagues [114]. They describe the various exploratory prcoedures (EP) that humans can perform to understand various object properties such as volume, temperature, friction, etc. Multi-modal learning is a key component for how biological agents learn and build models of objects. It can be argued by looking at failure modes to modern day robotics ([1]) that it is exactly this lack in multi-modal learning that requires further study.

Earlier work in haptic exploration includes ([28], [67]) who employed various hand engineered features to recognized objects using haptics. The challenges faced were largely due to robust sensors and the ability to control these sensors to explore objects effectively.

More recently, Chu et al. ([33]) measure various physical properties of objects using the bio-tac sensor using five different exploration procedures (EP). In addition, they also collect adjectives for each object and the corresponding They then compute precision, recall scores using a static hand-engineered feature and dynamic feature model employing Hidden Markov Models and compute precision, recall scores on a held out dataset. Similarly, [166] et al. also classify objects using a bag-of-words approach.

Romano et al. ([159]) mimic human tactile sensing for grasping by hand engineering features that can appropriately measure slippage. They then design a control strategy that can grasp and place that employs the tactile responses. They show that in cases where objects are crushable, a naive controller crushes 100% of the time as compared to a controller that effectively leverages tactile sensing.

Others, such as Sinapov et al. ([173]) have considered building object representations using vibrotactile sensation. They show that they can classify surface properties using data collected from five different EPs. Similarly, [54] classify textures using the bio-tac sensor using a Bayesian exploration strategy. While, [71] employ a *palpatation sequence* that is not learnt to effectively explore objects in a multi-fingered robot.

Our work relates to work by [58] who show that combining visual and haptic information can lead to better classification of haptic properties. More recently, Calandra et al. ([22]) show that employing tactile inputs into a learnt model can help improve predictions of graspability. ([144]) have shown that tactile features may not be required for certain constrained in-hand manipulation tasks. While this may seem contrary, this in fact is not a representative task. Further, the setup employed by the authors substitutes tactile sensing with a very rich 3D data along with a powerful learning method thus navigating around tactile sensing requirements.

Problem Setup

In our work, we pose the problem of object identification using curiosity based exploration in tactile space. What makes our problem further challenging is that the object to be

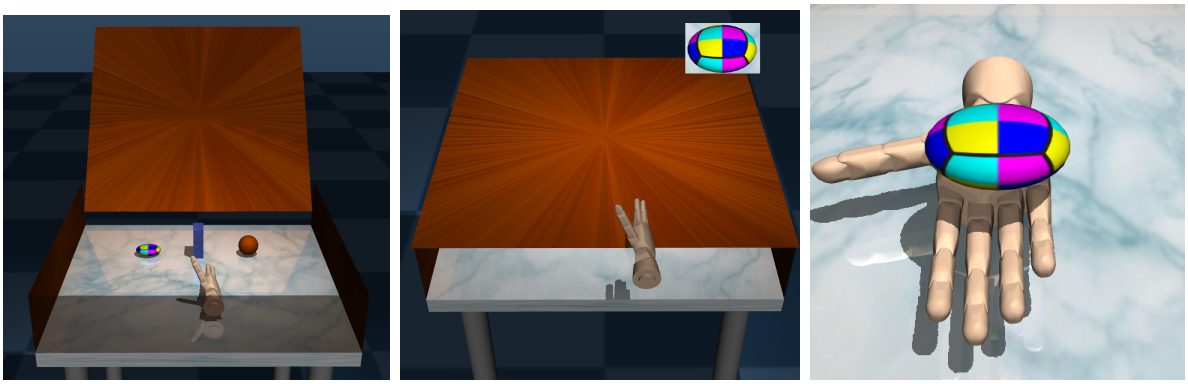


Figure 6.1: (*Left*) Shows our experimental setup. Three objects are in a drawer and a dexterous hand equipped with tactile sensing can explore novel objects using deterministic routines. (*Middle*) We are presented with a query image as seen by the inset in the top right of the image. We explore the objects in the drawer using tactile sensing only to identify the object (*Right*) We then retrieve the object by applying a grasping routine

identified is provided in *Image space* thus also requiring cross-modal learning. In the present version of the work we only consider the problem of exploration.

The agent is provided only with a visual image of the object to be retrieved, it must translate this into the representation space of tactile sensing to retrieve objects only by touching them. In the general case of retrieving the object, the agent must first explore spatially to locate where the objects are. Once it finds the object, it must move its fingers in an exploratory manner to collect information required to determine if the object is the one that needs to be retrieved. In the present work, we assume that all identification and classification is being done using only haptics. That is, no visual inputs are employed.

The tactile information is collected using a variety of different exploration policies. In our setup the agent learns a mapping from visual to tactile signals. This mapping enables the agent to determine the representation of the image in the representation space of tactile sensing (i.e. expected tactile response). The agent explores each object present in the drawer by touching it and compares the result of its exploration with the expected tactile response. Performing this comparisons requires a good representation of raw tactile signals.

6.1 Experimental Setup

Task Setup :

Figure 6.1 presents our task setup. A subset of objects from Figure 6.3 are placed in a drawer. The agent explores each object using one of the policies described further in the text. The agent then tries to discriminate the objects based on its tactile exploration. In our setup the movement between the objects and grasping, is done using a pre-determined

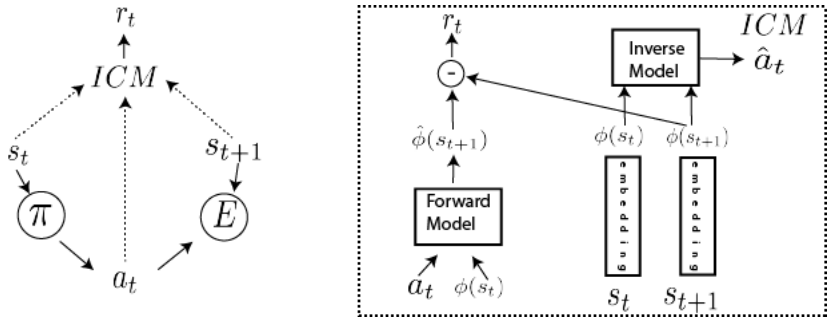


Figure 6.2: Intrinsic Curiosity Module (ICM): to step away from a pre-determined grasp we explore intrinsic curiosity as a way to generate a policy that helps explore each object presented. Essentially, ICM tries to predict the next time-step through the forward model but using an embedding space as opposed to raw feature space

routine.

The object is held translationally fixed in space but can rotate about its vertical (commonly called z) axes. The hand is initialized close to the object. The hand is translationally fixed, in that it cannot slide but it can rotate around the wrist joint. The fingers can explore their movements with only the restrictions imposed by the joints themselves. That is the fingers , say, cannot bend backwards towards the wrist.

For each episode of 500 time steps, the haptic forces $\mathcal{H}_t$ and the corresponding Images $\mathcal{I}_t$ are collected. Each object is presented in multiple random poses. The dataset consists of 500 haptics samples per object, each sample is averaged over 500 timesteps and has 19 dimensions. We divide the objects into a set of 10 training objects and 15 testing objects. We normalize both the images and the haptics vectors by setting image I_t to

$$\frac{I_t - 125.0}{255.0}$$

and haptics trajectory $\mathcal{H}_t$ to

$$\frac{\mathcal{H}_t - \mu_{\text{hapt}}}{\sigma_{\text{hapt}}}$$

Where μ_{hapt} and σ_{hapt} are the per feature sample mean and standard deviation of the haptics trajectories respectively. For each object we split the corresponding normalized 500 trajectories into 400 training trajectories, 50 validation trajectories and 50 testing trajectories.

We can test classification accuracy with two distinct testing routines that use embedded haptics features with a K nearest neighbors classifier (KNN) and the confidence values from a binary classifier.

During test time, we present the agent with a set of three objects (unseen during the training time). The goal given is to identify a single object from the set given an image. The agent explores each of the three objects a fixed number of times and the haptic forces for each exploration are stored. These haptic forces are then compared against the predicted

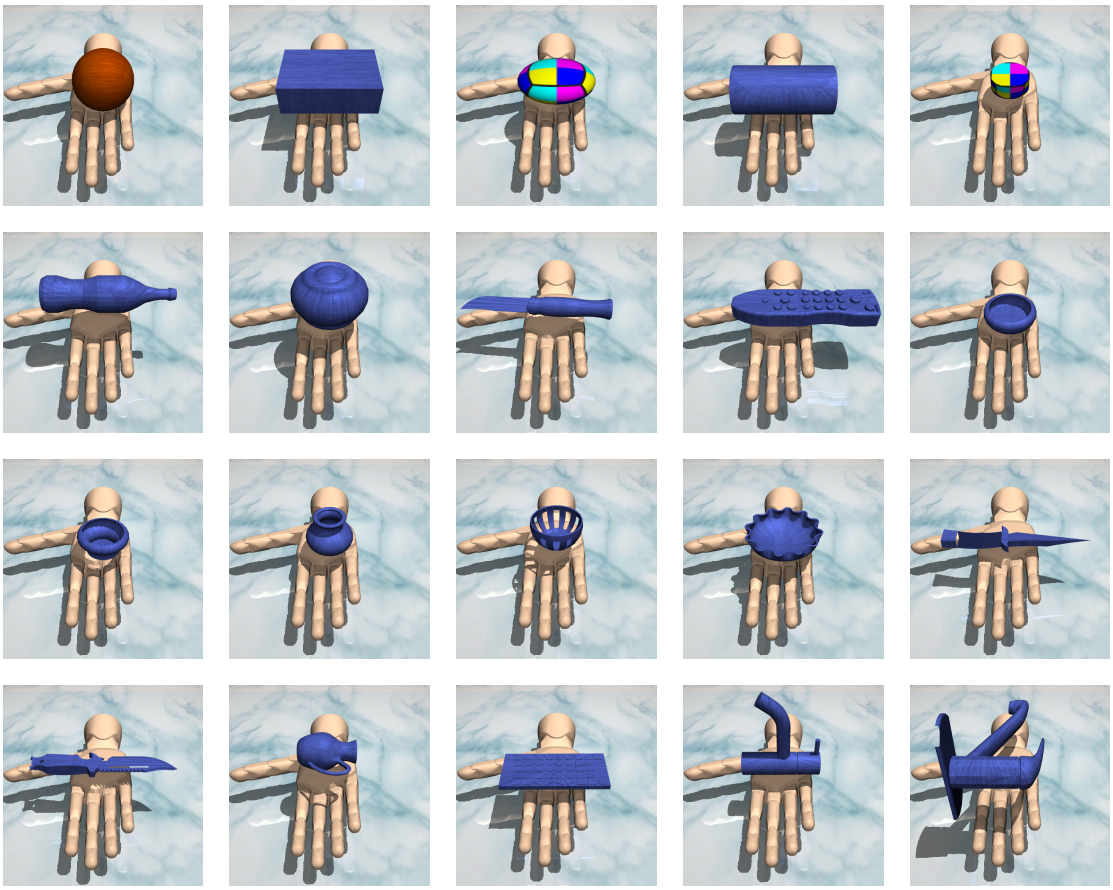


Figure 6.3: Displays the objects used in our experiments. We used a set of $\approx$ 45 objects. These were imported from the ShapeNet dataset ([29]). Each object was presented in various different poses. The hand was initialized at the same location for each sample while the object was randomized in each trial. Here we show some of the objects used

haptic response for the image and the test object that most closely matches the expected response is chosen. We refer to the average haptic response of an exploration as a trajectory.

Data Setup:

We use a simulated model of the anthropomorphic hand used as part of SouthHampton Hand Assesment Procedure (SHAP) test suite built by [123] (see Figure 6.4). The SHAP procedure was established for evaluating prosthetic hands and arms. With this idea in mind, prior work ([153]) built a prosthetic arm which could theoretically perform all useful human hand movements. Based on this hand and the DARPA Haptix challenge, a simulated model

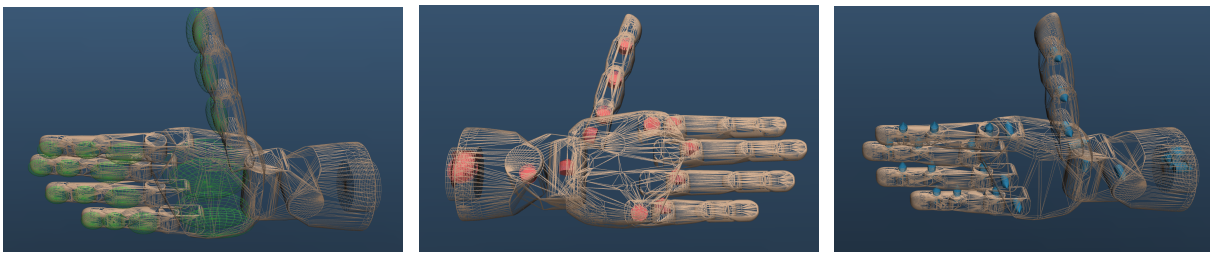


Figure 6.4: For fine manipulation humans rely mostly on touch, dexterous hands that are equipped with touch sensors could help mimic complex movements(*Left*) Shows the MPL hand with 19 touch sensors depicted in green. (*Middle*) The actuators can be seen in red. (*Right*) Shows the joints that are available in this hand. The hand is under-actuated so the number of joints are greater than the number of actuators

of a similar hand (but with fewer sensors) ([106]) was built using the *Mujoco* physics engine ([184]). This model was made publicly available and we use this for all our experiments.

The hand has five fingers and 22 joints out of which many are tendon coupled. For example, curling the tip of a finger automatically actuates the other two joints on the finger so that the finger moves towards the palm. Because of these couplings, the resultant dynamics can be quite complex and articulated. Out of the 22 joints, 13 are actuated. Out of these thirteen, ten joints control the motion of fingers and the other three control the rotation of the hand. Additionally, there are three degrees of motion along the (x, y, z) axis that are constrained in our problem setup and therefore our task must be optimized over 13 degrees of actuation. The hand is controlled by setting the position of these 13 actuators. In addition, the hand is equipped with 19 contact sensors (as seen in 6.4) that measure normal contact forces that are computed by *Mujoco*. These sensors form the basis of our tactile sensing.

6.2 Results

Data

For each object, haptic forces are collected using a policy. Thus the haptic forces corresponding to a trajectory and the object present in that trajectory constitute a data sample in our dataset.

Reinforcement Learning and Curiosity

We attempt to solve the task of classifying objects based on haptic feedback recieved during exploration. Doing this requires learning a meaningful exploration policy and using an effective method of classification from data sampled from a learned policy.

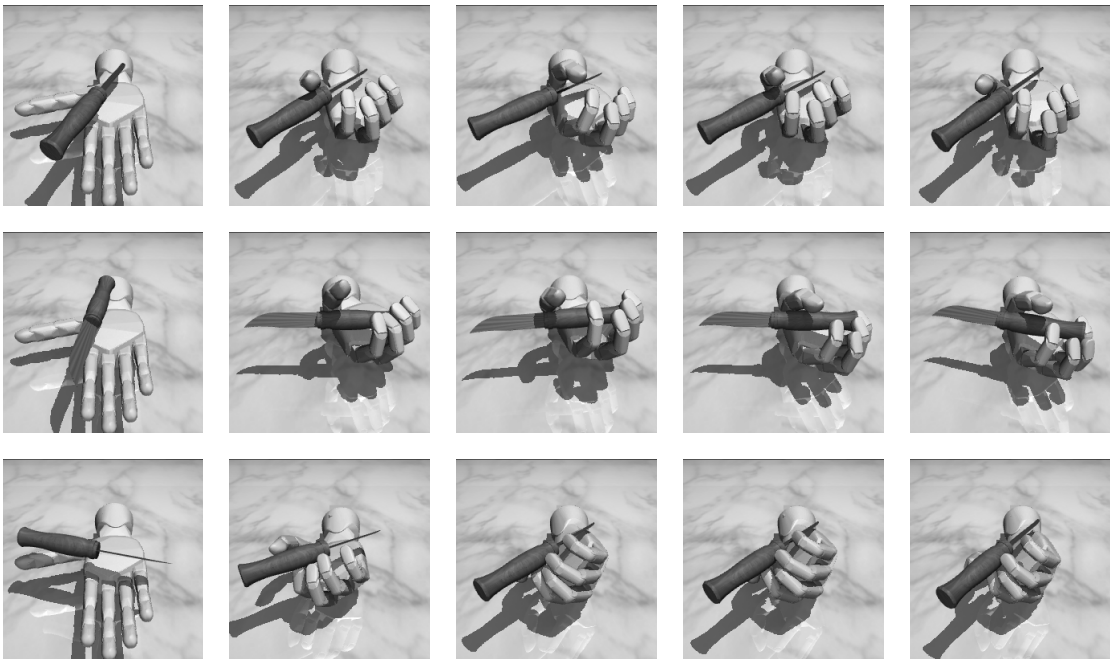


Figure 6.5: Displays sampling of a training object by each of the policy types: random (top), predetermined (middle) and curiosity driven reinforcment learning policy (botto). (Left to right) the policy unravels over time step 0 to time step T.

We use several distinct policies to evaluate different exploration strategies. First is a predetermined grasping routine, that slowly increases actuation force on the hand, closing the digits around the palm. This policy is not learned and will not show any object specific behavior.

Second, we learn an effective exploration policy with reinforcement learning.We evaluate generalization to 15 new objects with two approaches during test time. We have 500 trajectories per testing object, and partition

K Nearest Neighbors

We train a classifier that predicts object classes from haptic values. The network consists of three fully connected layers of dimension [250, 250, 10]. The first two layers use a relu activation and the final layer uses a softmax activation. To regularize we apply dropout to the second layer with rate 40%. We optimize the cross entropy loss for the 10 training objects

to solve the classification problem. We tested applying batch normalization to the first two layers, but due to prediction errors of the haptics regressor, predicted haptics have a different underlying distribution from the training data and generalization suffered considerably. We use the latent space of this classifier as an embedding of haptics features to use in testing.

Using this embedding we can define the KNN testing procedure:

1. Pick 3 objects from the 15 test objects.
2. Choose one to be the target object and select a trajectory from the target object's validation trajectories as the query point.
3. For each object in the 3 sample objects, randomly select 5 trajectories from the fitting trajectories and fit a kNN classifier with $k = 3$ to those haptic values.
4. Use this classifier to predict the class of the query point.

We define one test as 500 queries evaluated by the above procedure and can run multiple tests to compute the average and standard deviation in testing accuracy.

With this testing routine defined we can compare the goodness of the trained models. Take the layer prior to the classification layer of the trained classifier as a learned embedding of haptics features. We can test the difference in using predicted haptics from the regressor versus ground truth haptics by making predictions from the images for the test objects and embedding these predictions. We can then compare the KNN performance defined above on these embedded predicted values to the embedded ground truth haptics.

Extrinsic Reward for the Task of Classification

While a training an RL policy with no extrinsic reward signal should yield an effective exploration policy, it does not guarantee a policy that meaningfully solves the classification task. We introduce an extrinsic reward to the environment based on classification accuracy on the training objects. We construct an MLP classifier with the same architecture mentioned in the KNN section.

At the start of training, only an intrinsic reward is given to the agent. After 50 policy updates, the haptics signals generated during policy samples are saved as a series of 19 dimensional vectors in a buffer. Each element in the buffer is a [T x 19] array of haptics corresponding to a full episode sampled from the current policy.

The classifier is updated every 50 policy updates by averaging each element in the buffer to yield a training batch consisting of 19 dimensional trajectory averages. To maintain consistent normalization for the input data as the policy changes, a running mean and variance vector is updated by the statistics of the haptics buffer each time the classifier is updated.

After updating the classifier, a portion of saved trajectories is removed from the buffer to ensure that the distribution of the buffer does not drift away from the distribution of

trajectories being sampled from the current policy. This portion is decayed exponentially, begining at η_0 and after each update k, $\eta_k := \eta_0 * \eta_{k-1}$.

To compute extrinsic reward for a given timestep, the average haptics signal for the episode up to the current timestep is computed and fed through the classification network. The softmax value corresponding to the correct class label is used as the extrinsic reward, yielding a value between 0 and 1.

Policy Comparison

In this work, we explored three types of policy - (a) a random policy, (b) a pre-determined palpation, and (c) a curiosity based RL policy. The random policy samples forces from a normal distribution. The predetermined grasp is a hand coded solution that simulates a grasp-like palpation of the object. The curiosity policy is trained through an RL framework and is composed of 5 seperately trained policies each trained to sample a specific training object.

In our work we adapted and extended ICM [150] to continuous spaces and higher dimensional action spaces. Figure 6.2 presents a schematic of the curiosity model. We explored a wide range of intrinsic rewards – these are primarily tied to the observation spaces. In this work, we only present results from using vision (image of the object) as the observation. We have also experimented with other spaces such as state only, etc. For extrinsic rewards we explored negative regularized control ($-\|a\|^2$ as a way to choose actions that are somewhat smooth. We found that this choice led inevitably to movements being minimized. We discovered that if instead we used mean haptic force as the extrinsic reward, the curiosity model performed interesting exploration.

Table 6.1 shows a comparison of the three policies described above using the KNN testing procedure and the binary classification procedure. We compare the accuracy when using haptics predictions (Predicted) versus ground truth haptics. While we see grasping behavior that changes based on the given object in our curious policy, the curious policy performs worse than our predetermined policy in KNN tests but outperforms random.

We found that batch normalization greatly improved the accuracy of the network to converge for the curious policy data. We also tested the effects of training the networks on smaller chunks of time (across columns in the tables). We find that in general chunking the episode into approximately 10 to 20 chunks really helped improve performance.

We also tested the effects of varying the number of training objects for the various policies. These can be seen in Table 6.3. We find that in general, more training data helped the performance of all policies.Finally, we tested how these policies generalized to novel objects. These results can be seen in Table 6.2. We find that both the predetermined and curious policies generalize well.

Policy	1 Interval	5 Intervals	10 Intervals	20 Intervals
Predet.	80.7% ± 0.6%	84.0% ± 0.2%	85.2% ± 0.2%	84.9% ± 0.4%
Hptx + Class Rew	80.5% ± 1.5%	82.6% ± 1.3%	83.8% ± 1.4%	85.8% ± 1.1%
Hptx(Curious)	77.9% ± 0.7%	79.8% ± 0.3%	81.9% ± 1.1%	82.8% ± 0.8%
St.+Hptx(Curious)	70.7% ± 4.5%	74.2% ± 3.2%	76.1% ± 2.0%	77.7% ± 1.4%
Vision(Curious)	73.4% ± 1.4%	73.5% ± 1.9%	73.6% ± 2.0%	74.6% ± 2.1%

Table 6.1: Testing accuracies for curious policies using different observation types. Policies were trained with no extrinsic reward signal. Full episodes were then sampled to gather haptics values, each episode being 500 simulation steps. Those 500 steps were then divided evenly into sequences which were averaged across. For example 5 sequences would divide each 500 timestep sequence into 5 chunks of 100 steps, resulting in an averaged haptics vector of dimension 5*19. These were then evaluated using the KNN testing procedure as described in section 3.1. The reported accuracy is an average across 3 seeds. The first row shows results for a predetermined policy. The second row shows results for an observation space that has only haptics for intrinsic curiosity but also the class rewarder as the extrinsic accuracy. The third row shows results for haptics as a feature space using intrinsic curiosity. The fourth row combines both state information and haptics for intrinsic curiosity. The last row shows vision as the observation to the intrinsic curiosity module

Policy	#Obj	1 Interval	5 Intervals	10 Intervals	20 Intervals
Predet.	15	80.7% ± 0.6%	84.0% ± 0.2%	85.2% ± 0.2%	84.9% ± 0.4%
	25	79.9% ± 1.1%	84.5 % ± 0.5%	84.5 ± 0.3 %	83.7 +- 0.5 %
	35	78.5% ± 0.4%	82.9% ± 0.3%	83.9 % ± 0.5 %	83.2 % ± 0.0%
Haptics (Curious)	15	80.2% ± 2.5%	80.6% ± 2.3%	81.9% ± 2.6%	84.6% ± 1.8%
	25	80.1% ± 3.3%	80.6% ± 2.8%	82.7% ± 2.2%	83.8% ± 1.0%
	35	80.7% ± 2.0%	80.4% ± 1.6%	82.0% ± 1.0%	83.6% ± 1.1%

Table 6.2: Testing accuracies for different policies using a different number of **test objects**. Policies were trained with no extrinsic reward signal. Full episodes were then sampled to gather haptics values, each episode being 500 simulation steps. Those 500 steps were then divided evenly into sequences which were averaged across. For example 5 sequences would divide each 500 timestep sequence into 5 chunks of 100 steps, resulting in an averaged haptics vector of dimension 5*19. These were then evaluated using the KNN testing procedure as described in section 3.1. The reported accuracy is an average across 3 seeds.

Policy	#Obj	1 Interval	5 Intervals	10 Intervals	20 Intervals
Haptics (Curious)	5	77.2% ± 1.9%	76.3% ± 0.2%	77.3% ± 0.5%	78.0% ± 0.8%
	10	77.9% ± 0.7%	79.8% ± 0.3%	81.9% ± 1.1%	82.8% ± 0.8%
	20	78.5% ± 1.3%	81.6% ± 1.9%	82.5% ± 2.0%	85.1% ± 2.2%
	30	79.8% ± 0.9%	83.8% ± 0.9%	85.4% ± 1.3%	86.5% ± 1.7%
Haptics (Class Rew)	5	72.5% ± 3.3%	72.5% ± 2.9%	75.4% ± 1.9%	74.6% ± 3.6%
	10	80.5% ± 1.5%	82.6% ± 1.3%	83.8% ± 1.4%	85.8% ± 1.1%
Predet.	5	79.4% ± 0.4%	80.8% ± 0.2%	82.4% ± 0.2%	82.2% ± 0.6%
	10	80.7% ± 0.6%	84.0% ± 0.2%	85.2% ± 0.2%	84.9% ± 0.4%
	20	83.4% ± 0.2%	86.8% ± 0.6%	87.2% ± 1.0%	87.3% ± 0.4%
	30	84.5% ± 0.8%	87.1% ± 0.5%	86.5% +- 0.1%	88.0% ± 0.0%

Table 6.3: Testing accuracies for curious policies using smaller and larger sets of **training objects** with ICM batch normalization disabled. The class reward policy was trained with intrinsic and extrinsic reward, whereas the curious policy was trained with a purely intrinsic reward. were trained with no extrinsic reward signal. Full episodes were then sampled to gather haptics values, each episode being 500 simulation steps. Those 500 steps were then divided evenly into intervals which were averaged across. For example 5 intervals would divide each interval timestep sequence into 5 chunks of 100 steps, resulting in an averaged haptics vector of dimension 5*19. These were then evaluated using the KNN testing procedure as described in section 3.1. The reported accuracy is an average across 3 seeds.

6.3 Discussion

Here, we present preliminary results of a curiosity driven exploration policy using tactile sensing as observation for object discrimination. We find that the curious policy does perform interesting behaviors as seen in Figure 6.5 and also classifies the novel test objects fairly well. While the curiosity driven policy does perform interesting behavior the classification accuracies do not significantly outperform a simple palpation based predetermined policy. This is in part because the task setup may be too trivial and also the extrinsic reward (goal directed reward) could be better specified to improve performance.Future work in these directions could be very interesting for the field.

An interesting direction to pursue is to vary the type of observations the RL policy and the curiosity module accept. For example, the policy observation could have vision for inputs and the curiosity observations could have haptics as inputs. This could lead to more interesting scenarios that one may consider.

6.4 Acknowledgements

The work in this chapter was performed in collaboration with Blake Tickell and advise from Pulkit Agrawal and Jitendra Malik. This is the first presentation of this work anywhere.

Chapter 7

Manipulation by Feel: Touch-Based Control with Deep Predictive Models

Touch sensing is widely acknowledged to be important for dexterous robotic manipulation, but exploiting tactile sensing for continuous, non-prehensile manipulation is challenging. General purpose control techniques that are able to effectively leverage tactile sensing as well as accurate physics models of contacts and forces remain largely elusive, and it is unclear how to even specify a desired behavior in terms of tactile percepts. In this paper, we take a step towards addressing these issues by combining high-resolution tactile sensing with data-driven modeling using deep neural network dynamics models. We propose deep tactile MPC, a framework for learning to perform tactile servoing from raw tactile sensor inputs, without manual supervision. We show that this method enables a robot equipped with a GelSight-style tactile sensor to manipulate a ball, analog stick, and 20-sided die, learning from unsupervised autonomous interaction and then using the learned tactile predictive model to reposition each object to user-specified configurations, indicated by a goal tactile reading. Videos, visualizations and the code are available here:
`https://sites.google.com/view/deeptactilempc`

7.1 Introduction

Imagine picking up a match stick and striking it against a matchbox to light it, a task you have performed with ease many times in your life. But this time, your hand is numb. In 2009, Johansson et al. [96] performed this experiment, studying the impact of anesthetizing the fingertips of human subjects on their ability to perform this task. The results were striking: human subjects could barely manage to pick up a match stick, let alone light it. Videos of this experiment show human clumsiness [95] that is strikingly reminiscent of the faltering, lurching struggles of modern robots [1].

Why did taking away the sensation of touch have such an impact on these subjects? Touch is unique among sensory modalities in that it is physically immediate and permits direct measurement of ongoing contact forces during object interactions, from which it is possible to infer friction, compliance, mass, and other physical properties of surfaces and objects. This knowledge is critical for manipulation tasks like matchstick striking. Visual sensing is a poor substitute: not only is it physically remote, but it is also

Figure 7.1: (*Left*) For fine manipulation, humans rely mostly on touch, as vision is occluded by the finger itself. (*Right*) Our custom-built GelSight touch sensor. We train a video prediction model on the tactile modality, and use this model to perform object repositioning tasks.

usually occluded by the actuators at the points of contact. Manipulation without touch is perhaps akin to navigation without vision.

The importance of tactile sensing has long been acknowledged in the robotics community [198, 40, 78], but exploiting touch in robots has proven exceedingly challenging for three key reasons: (i) tactile sensing technology has largely been limited to sparse pointwise force measurements, a far cry from the rich tactile feedback of biological skin, (ii) accurate physics models of contacts and forces have remained elusive, and (iii) it is unclear how to even specify a desired tactile goal, such as holding a matchstick, or striking it obliquely against the matchbox.

In this paper, we show the promise of robotic control with rich tactile feedback, by exploiting recent advances to tackle these difficulties. Firstly, deformable elastomer-based tactile sensing (e.g., GelSight) has proven to be an extremely versatile, high-bandwidth, high-spatial resolution alternative to traditional tactile sensing [199, 22]. This provides our first stepping stone.

Secondly, while high-resolution touch sensing such as GelSight (see 7.1 and 7.2) provides us with adequate sensory signals, we also need a control mechanism that can handle such high-dimensional observations and choose intelligent actions [136, 118]. Our second stepping stone comes from the deep learning literature, where deep models of high-dimensional data have been successfully employed to enable a variety of vision-based robotic skills [53, 116, 48].

Finally, as we will show, the combination of these two advances makes it feasible to plan towards tactile goals specified directly in the raw tactile observation space. Goal specification in this manner is not only much more informative than, say, in the space of forces at sparse contact points, but is also often much more natural for a user to specify.

Concretely, our contributions are as follows. We train deep model-based control policies that operate directly on observed raw high-dimensional tactile sensing maps. We show that such policies may be learned entirely without rewards, through diverse unsupervised exploratory interactions with the environment. Fi-

nally, we show how the desired manipulation outcomes for our policies may be specified as goals directly in the tactile observation space. We demonstrate and evaluate these contributions on a high-precision tactile ball rolling task, a joy-stick re-positioning task and a die rolling task: a robot arm with three linear axes is equipped with a tactile sensor at its end effector. Its goal is to move the end-effector into a configuration so that a desired tactile goal-pattern is measured with the sensor.

These tasks are designed to exhibit two key difficulties that are shared by a wide range of manipulation tasks: (i) An external object must be in contact with the robot end-effector in a specific desired configuration. This is a common feature of many manipulation and tool use problems. For example, when trying to light a match, a specific sequence of contact states needs to be attained. (ii) While performing the task, the object of interest, such as the ball bearing, becomes occluded from view, therefore controllers or policies that only have access to a remote visual observation are unable to solve the task. In the experiments for these three distinct manipulation tasks, our method outperforms hand-designed baselines for each task. We see these results as an important step towards integrating touch into solutions for general robotic manipulation tasks.

7.2 Related Work

Prior work on touch-based control has proposed methods ranging from manual design of control laws [172] to extracting and controlling high-level features from touch sensors [121, 117]. In contrast to these methods, our approach does not rely on pre-specified control laws and features. We learn a general-purpose predictive model that can be used to accomplish a variety of tasks at test time. Furthermore, we use a high-resolution touch sensor based on the GelSight design [97], which provides detailed observations of the contact surface in the form of a camera image.

Prior work has also used reinforcement learning to learn to stabilize an object with touch sensing [84] and explored learning forward predictive models for touch sensors [186]. While this prior work used low-dimensional readings from a BioTac sensor, our model operates directly on the raw observations of a GelSight sensor, which in a our case is an RBG image downsampled to 48x64 pixels. We demonstrate that comparatively high-resolution tactile sensing in conjunction with the proposed tactile MPC algorithm allows to reposition freely-moving objects according to user-specified goals, a more complex task than those demonstrated in prior work [84, 186]. To address this high-dimensional prediction problem, we build on recent work on control via video prediction [53, 48]. Prior work in this area has used video prediction in combination with model-predictive control to perform non-prehensile object repositioning from RGB camera images. To our knowledge, no prior work has used video prediction models together with touch sensing for touch-based object repositioning. Concurrent work [183] learned a two dimensional latent space and dynamics model to perform control for following human demonstrations given in the tactile space, however handling of objects has not been shown yet.

A variety of different touch sensor designs have been proposed in the literature [198], though affordability, sensitivity, and resolution all remain major challenges. The BioTac [56, 55] sensor has been widely used in robotics research, particularly for grasping [31, 130], but it provides only a limited number of measuring channels (i.e., 19 or 22 for different configurations). The GelSight design, which consists of a camera that observes deformations in a gel, offers good resolution, though at the cost of latency [97]. In our case, this tradeoff is worthwhile, because the resolution of the sensor allows us to precisely reposition objects in the finger. GelSight-style sensors have been used in a number of prior works for other applications, including tracking [92], inserting USB connectors [122], estimating object hardness [200], and grasping [21]. To our knowledge, our work is the first to employ them for object repositioning with learned predictive models.

Figure 7.2: We evaluate deep tactile MPC on 3 different fine-grained manipulation tasks: (left to right) ball repositioning, joystick deflection, and die rolling to reach a specified face.

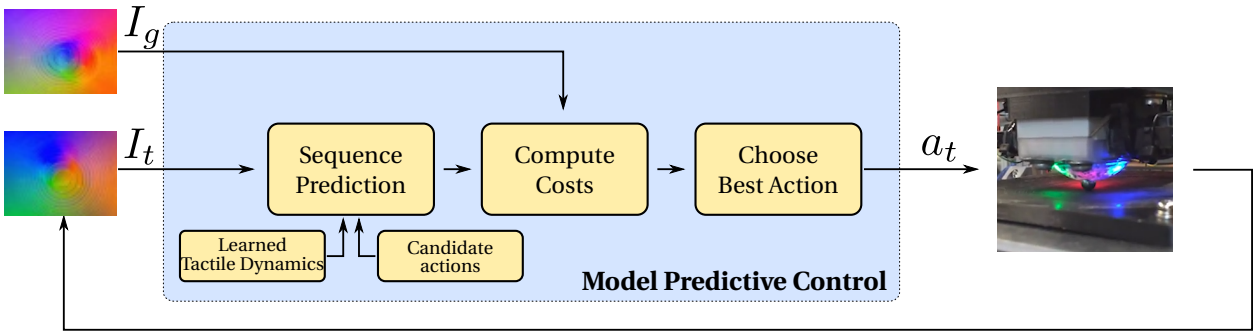


Figure 7.3: Deep tactile model predictive control: given the current tactile observation and a learned deep predictive model, we can predict the outcomes for different action sequences. We use this model within a model predictive control algorithm based on stochastic optimization. At each time-step the algorithm samples multiple potential action sequences and computes their cost, which depends on the difference between the predicted tactile observations and the goal observation. The first action of the actions sequence that attained lowest cost is then applied to the actuators.

7.3 Tasks and Hardware Setup

While our aim is to develop control approaches for general manipulation, we focus on three representative tactile control tasks: rolling a ball to a specified position, manipulating an analog stick from a game controller, and rolling a die to a specified face, see 7.2. Each of these tasks presents unique challenges, making them well-suited for evaluating our approach. In this section, we describe the tasks, the sensor, and the experimental hardware setup.

Ball repositioning task The ball repositioning task requires the robot to move a ball bearing to a target location on the sensor, which requires careful modulation of the contact force between the ball and the finger without the ball slipping out of the finger. Since the ball is underactuated, properly balancing the forces is critical.

Analog stick deflection task The second task requires the robot to deflect an analog stick. This task presents an additional challenge: the robot must intentionally break and reestablish contact to deflect the analog stick in the desired direction, since sliding the finger along the stick will deflect it in undesirable ways. We encourage the reader to view the supplementary video for an illustration. The only way to perform the task successfully is to lift the finger off of the stick by moving vertically, repositioning it, bringing it back

down onto the stick, and then deflecting the stick in the desired direction. This adds the difficulty of making the system only partially observable (when the finger loses contact with the joystick), and requires the model to acquire a more fine grained understanding of contact dynamics.

Rolling a 20-sided die This third task requires the robot to roll a 20-sided die so that a specified die face is facing upwards. This task was chosen for two reasons: First this task has a comparably high-level of difficulty due to slippage and undesired rolling of the die. Second this task allows us to use a simple and intuitive success metric — the fraction of trails where the desired face ended up on top.

Tactile sensor For tactile sensing, we use a custom elastomer-based sensor based on the GelSight design [199] with a diameter of $4cm$. A standard webcam is embedded directly into the Gel producing high-resolution images of the surface deformations of the elastomer. Example images are shown in 7.4. Our choice of tactile sensor is key to our approach for the two reasons: (i) it allows us to use comparatively high-resolution observations, which aids both in control, as well as in setting expressive self-supervised goals at test time (as we will show), and (ii) it naturally offers a compliant surface at the point of contact, which is important in many manipulation settings including ours.

Hardware setup In order to study tactile control with our sensor, we mount it on a modified miniature 3-axis CNC machine (see 7.10 in the appedix). This machine has a precision of $0.04mm$, which allows it to reposition the sensor accurately based on the actions commanded by our controller.

Autonomous data collection To train our deep predictive model, we need to collect training data of the robot interacting with its environment. We autonomously collected 7400 trajectories for the ball, around 3000 trajectories for the analog stick, and 4500 trajectories for the die experiment. Each trajectory consists of 15 to 18 time steps, depending on the experiment. These training trajectories were collected by applying random movements along each of the three axes. More details about the data collection process are provided in 7.8.

7.4 Deep Tactile Model-Predictive Control

The use of a high-resolution sensor such as the GelSight enables fine control, but also presents a major challenge: the high-dimensional observation space makes modeling and control substantially more difficult. To allow performing a variety of different manipulation tasks at test-time using a large dataset collected beforehand, we explore a model-based method for touch-based control. Our method builds on prior work on control via visual prediction [53, 48]. In this class of methods, a deep recurrent convolutional network is trained to predict future video frames conditioned on the most recent observations and a sequence of future actions. More information about the model are provided in 7.8. At test time, this model can be used to perform a variety of manipulation tasks by optimizing over the actions until the model produces the predictions that agree with the user's goal, and then executing those actions on the robot. Prior work has applied this approach to vision-based object repositioning [53, 48].

Deep predictive model The particular model that we use has the same architecture as the video-prediction models proposed in prior work [53, 48]. Concretely, we use the architecture proposed in [116], and train it to predict future GelSight sensor observations $\hat{I}_{1:T} \in \mathbb{R}^{T \times H \times W \times 3}$ conditioned on the current observation I_0 and a sequence of candidate actions $a_{1:T}$, where T denotes the prediction horizon.[1] This predictive model can be written as $\hat{I}_{1:T} = g(a_{1:T}, I_0)$. In Figure 7.4 we show several example predictions of our model on test-set trajectories. We can see that the model accurately predicts the contact pattern for a sequence of 13 time-steps into the future.

Goal specification At test-time, the user specifies a goal by providing a goal *tactile image*: a reading from the GelSight sensor for the desired configuration, which we denote as I_g. While a number of methods

[1] While the architecture was proposed to generate stochastic predictions in [116], we train a deterministic variant instead.

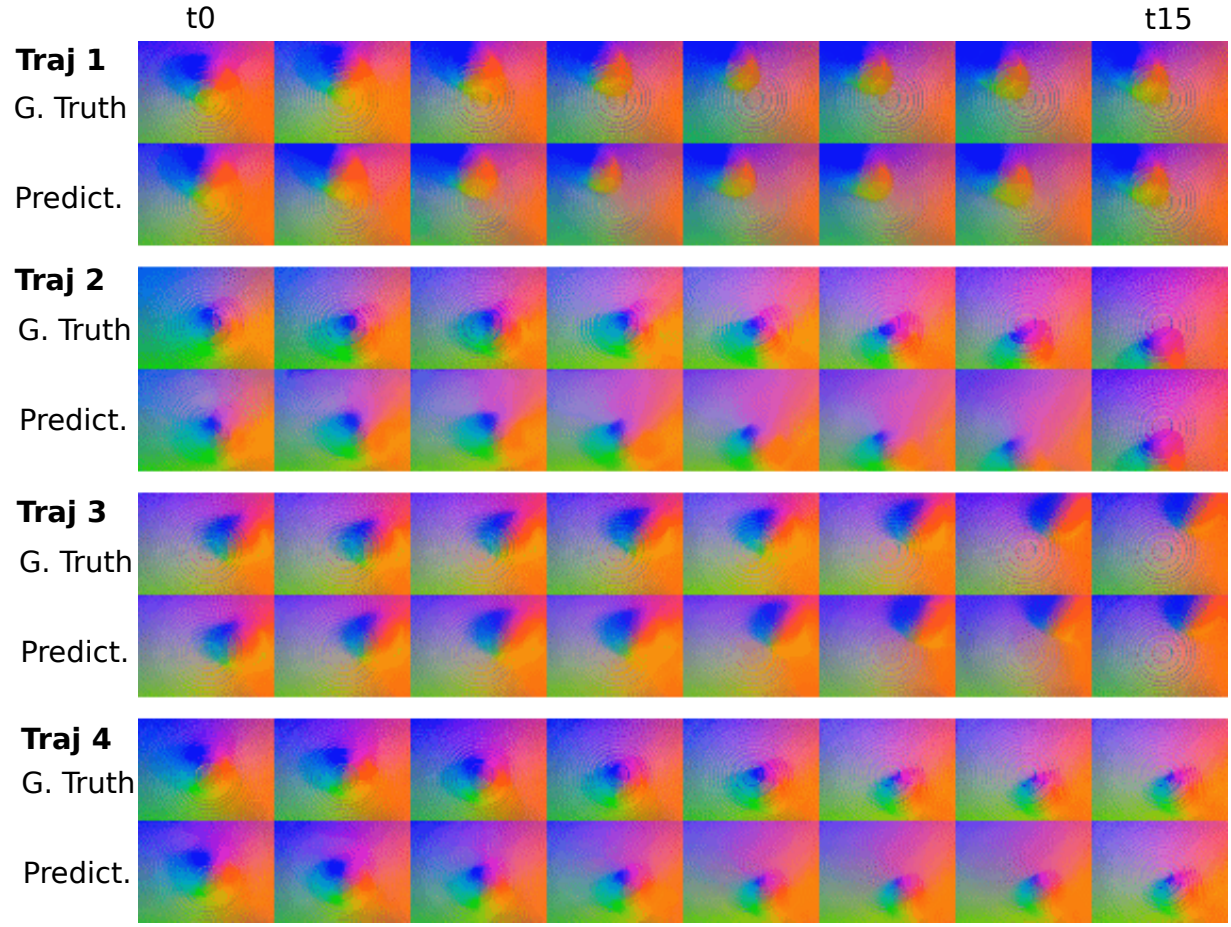


Figure 7.4: Four different predicted sequences for the ball bearing task, conditioned on images and actions from the test set: the top film strip in each row shows the ground truth observations, the bottom one shows the predictions made by our model when conditioned on the same action sequence. The actions consist of movements between 0 and 2.8mm in length along the horizontal axes, and between 0 and 0.4mm in length along the vertical axis.

could be used to specify goals, this approach is simple and general, and allows us to evaluate our method on a "tactile servoing" task.

Tactile MPC control Once the predictive model has been trained, we may use it to plan to achieve any user-specified goal configuration I_g in the tactile observation space. For this, we employ model-predictive control with the learned predictive model. We use an optimization-based planner to optimize over the action sequence at each time step to determine the actions for which the predicted outcome is closest to goal tactile image I_g, as illustrated in 7.3. The planning problem is formulated as the minimization of a cost function $c_t(I_g, \hat{I}_t)$ which provides a distance metric between the predicted image $\hat{I}_t$ and the goal image I_g. In this work we set $c(\cdot, \cdot)$ to the mean squared error (MSE) in pixel-space between I_g and $\hat{I}_t$, such that the optimization is given by

$$a_{1:T} = \arg\min_{a_{1:T}} \sum_{t=1,\ldots,T} c_t(I_g, \hat{I}_t)\,, \tag{7.1}$$

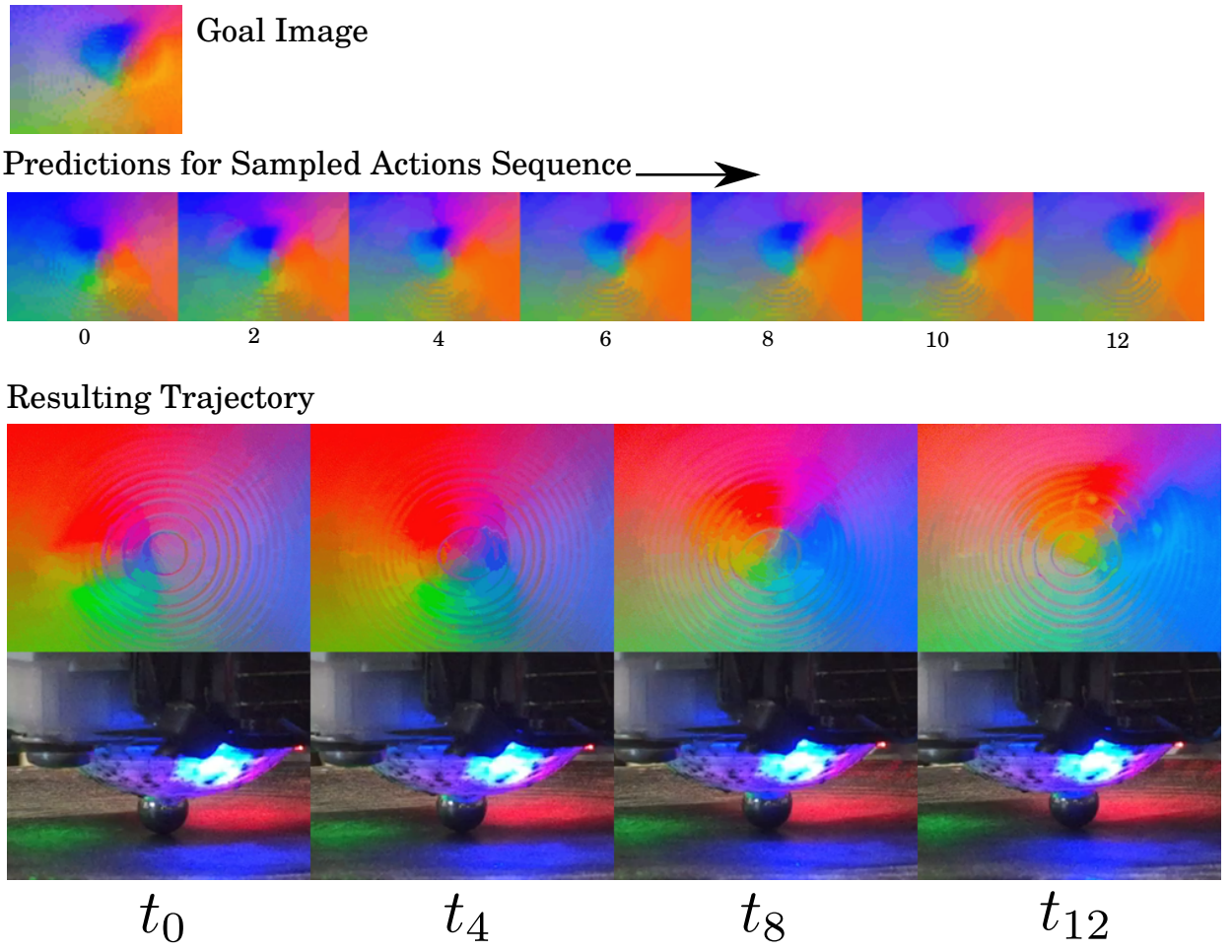


Figure 7.5: Example rollout for the ball-bearing task. The goal is to reach the the goal-image in the top row. The predicted frames for the action sequence that achieved lowest cost is shown in the second row, only every second prediction step is shown. The third row shows the actual trajectory taken by the robot for both the tactile image and side image.

where $c_t \in \mathbb{R}$. We perform sampling-based planning using the cross-entropy method (CEM) [161]. To compensate for inaccuracies in the model, the action sequences are recomputed at each time step $t \in \{0, ..., t_{max}\}$ following the framework of model-predictive control (MPC). At each real-world step t, the first action of the best action sequence is executed on the robot. The planning process is illustrated in 7.3. 7.5 shows the executing of tactile MPC on the ball-bearing task.

Implementation details We use three CEM iterations for optimization, with 100 samples each. The prediction horizon for the video-prediction model is between 15 and 18 depending on the task. Each action is repeated three times, such that the plan consists of five or six actions. This planning horizon is usually sufficient to reach the goal from most configurations considered in our experiments. Using time-correlated actions helps to reduce the search space for the optimizer, thus reducing the number of required samples and increasing the control rate.

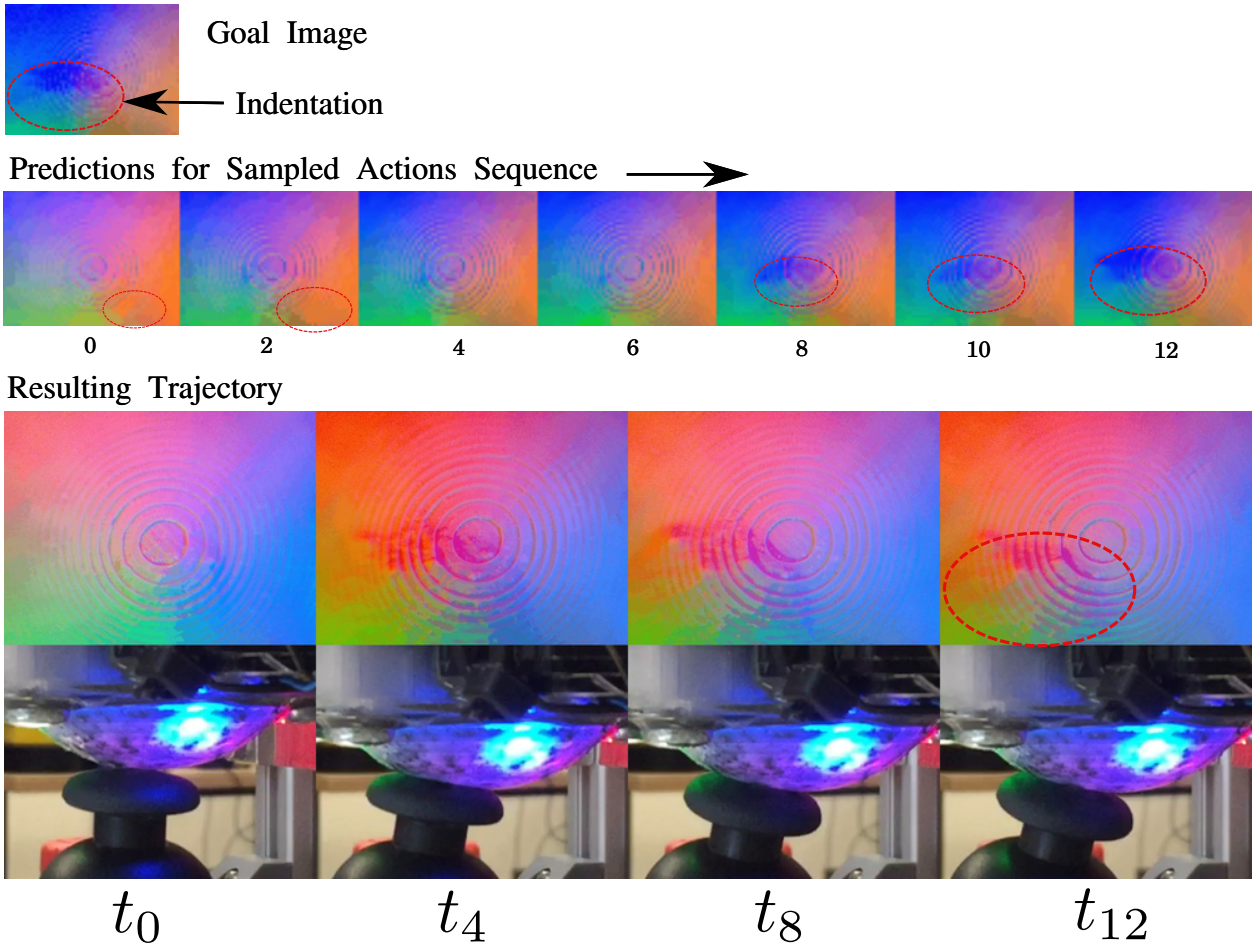


Figure 7.6: Example of successful analog stick tactile positioning task. In the second row we show the predicted images (every 2nd time-step) for the optimal action sequence found at (real-world) timestep 1. For the 1st timestep (second row) the pressure center is in the bottom right of the image indicated by a red ellipse, it then lifts off for several timesteps and comes back in the last five steps. The last image of the predicted sequences closely resembles the desired indentation shown in the goal image.

7.5 Experimental Results

We now experimentally validate our deep tactile MPC approach on three real-world tactile manipulation tasks: moving a ball bearing, positioning an analog joystick, and rolling a die. For video results, see the project webpage[2].

[2] https://sites.google.com/view/deeptactilempc

Evaluation Metrics

Evaluating the performance of the tactile policy is challenging, since the target is provided directly in the space of tactile observations, and the ground truth pose of the object is unknown. However the advantage of using tactile images as goals is that this is highly general, since a user can easily specify any goal by manually positioning the sensor in the desired configuration.

For our evaluation, we use three different metrics that quantify different aspects of control performance: 1) mean squared error (MSE) of the difference between the goal-image I_g and the tactile-image I_t observed at the end of an episode, and 2) manually annotated distance in pixel-space between the pressure centroid of the object at the end of the trjectory and the location of the pressure centroid in the goal image. 3) For the die-rolling task we have a more intuitively meaningful success metric — the fraction of trials in which the die could be rolled so that it has the desired face on top.

While the MSE metric can be automatically evaluated and exactly captures the objective we optimize for in tactile MPC, mean squared errors in image space do not necessarily reflect actual distances between object positions and poses. It is for this reason that we use the additional manually annotated distance measure.

Tactile Control Baseline

In order to provide a comparative baseline for our tactile MPC method, we designed an alternative method that uses hand-engineered image features to solve each of our three evaluation tasks. It accomplishes this by first detecting the pressure centre in the imprint of the ball, joy-stick or die and then moving in a straight line towards the target position. To estimate the coordinates of the pressure center in the current image and in the goal image I_g, it computes the weighted centroid of the pixels in the current image, where weights are squared pointwise differences between the current image and a blank "background" image from the sensor when it is not in contact with any object. This difference image outlines the contact region, and detecting its centroid roughly localizes the object of interest.

Having detected an approximate position of the object in both the current and goal image, the baseline commands an action along the vector from the current estimated contact position to its estimated position in the goal image. The step length is tuned to achieve the maximum control performance in terms of the estimated distance between the current and desired object position. Note that this is a fairly strong baseline for the ball bearing task, since localizing the centroid provides a good indication for the location of the spherical ball. For the joystick and die-rolling task, this baseline fails frequently, yet it is hard to design a better baseline. In contrast, deep tactile MPC is more general and does not require manual tuning or knowledge about specific object mechanics. The deep dynamics model used in tactile MPC learns a variety of basic properties about the world purely from data – such as that ball bearings remain in one piece and move opposite to the direction of movement of the sensor.

Manipulating a Ball, an Analog Stick, and a 20-sided Die

We find that our method enables a robot to achieve all three manipulation tasks through only touch sensing, without other sensory feedback. For example, the robot is able to maneuver a ball, manipulate a die to new faces, control a joystick, all entirely by feel. For qualitative examples of these results, see Figures 7.5, 7.6 and 7.9, as well as the supplementary video[3].

For the ball repositioning task, on the left of 7.7 we show a plot that illustrates the fraction out of 30 test trajectories that obtained distances between the pressure centroid at the final time-step and goal image which are lower than a certain threshold. The positions were hand-labeled, the distances are measured in terms of distances in the 64x48 tactile image. The right sight of 7.7 shows the same graph for the mean-squared error

[3] `https://sites.google.com/view/deeptactilempc`

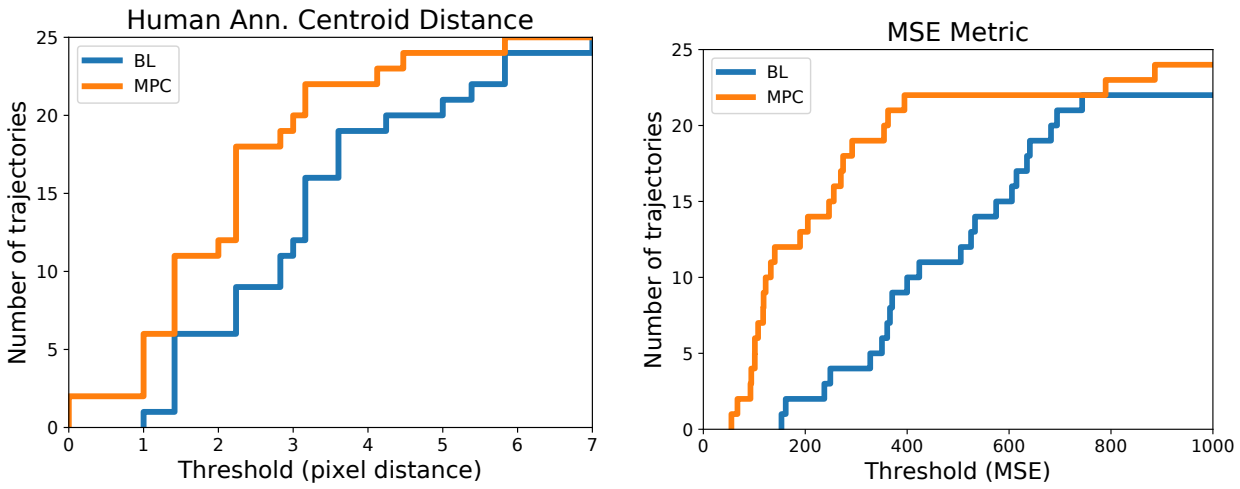


Figure 7.7: Quantitative analysis for ball task. (*Left*) The y axis shows the number of trajectories out of 30 total for which the pixel distance between the final and the goal position of the pressure centroid, as annotated by a human labeler, is lower than the threshold given by the x-axis. (*Right*) Number of trajectories with MSE distance to goal-image lower than threshold. A vertical slice through this can be used to determine what fraction of trajectories reach the goal within a certain distance. In all cases, our deep tactile MPC method (in orange) outperforms the hand-designed baseline approach (in blue) significantly.

between the final image and the goal-image. Note that for both metrics our method (in orange) consistently dominates the baseline approach (in blue) by a substantial margin.

The results for the analog stick repositioning task are shown in 7.8, using the same metrics. Again, we see that our method (in orange) substantially outperforms the baseline.

As shown in 7.1, our deep tactile MPC method achieves a significantly lower median distance than the baseline in both the ball-rolling and analog-stick task. In the die rolling experiments the difference between tactile MPC and the baseline is even larger. We conjecture that this is because of the fact that the dynamics in this die rolling task are too complex to be handled well by a simple hand-tuned controller, since it involves complex slipping, sliding and rolling motions on multiple surfaces. 7.9 shows a qualitative example (more in supplementary video).

Based on these results we conclude that using sampling-based planning in a combination with a deep

	Median L2 dist [mm]		Success Rate
	Ball Rolling	Analog Stick	Die
Tactile MPC	**2.10**	**5.31**	**86.6% (26/30)**
Centroid Baseline	2.97	8.86	46.6% (14/30)

Table 7.1: Benchmark results for the ball-rolling, analog-stick and die-rolling experiments. The median L2 distances are between the hand-annotated pressure centroid of the final and goal-image. For the die experiment we measure the fraction of examples where the desired face lands on top. Benchmarks are performed with 30 examples.

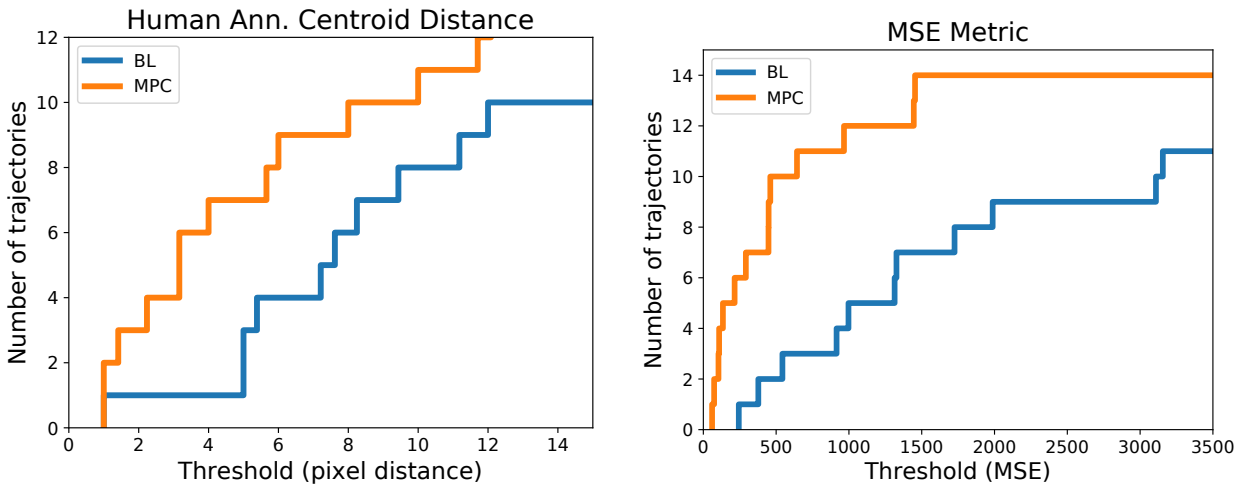


Figure 7.8: Quantitative analysis for analog-stick task. (*Left*) Number of trajectories, out of 15 trials, for which euclidean distance between the final position of the pressure centroid and the goal position, as labeled by a human labeler. (*Right*) Number of trajectories for which mean squared error (MSE) between the final image and the goal image is lower than threshold.

dynamics model is powerful method for solving a range of challenging manipulation tasks solely based on tactile information. We expect that the gap between hand-design methods and learning-based method will be even greater for more complex robotic manipulation scenarios such as multi-fingered manipulation.

7.6 Discussion and Future Work

Precise in-hand manipulation in humans heavily relies on tactile sensing. Integrating effective touch sensing into robotic control can enable substantially more dexterous robotic manipulation, even under visual occlusion and for small objects that are otherwise difficult to perceive. In this paper, we presented a touch-based control method based on learning forward predictive models for high-bandwidth GelSight touch sensors. Our method can enable a robotic finger to reposition objects and reach user-specified goals.

While our results indicate that deep convolutional recurrent models can effectively model future touch readings conditioned on a robot's actions, our method still has a number of limitations. First, we explore short-horizon control, where the goal can be reached using only tens of time steps. While this is effective for simple servoing tasks, it becomes limiting when tasks require rearranging multiple objects or repeatedly executing more complex finger gaits. However, as video prediction models improve, we would expect our method to improve also, and to be able to accommodate more complex tasks.

Another limitation of our work is that, with a single finger, the range of manipulation behaviors that can be executed is limited to simple rearrangement. A more dexterous arm or a multi-fingered hand could perform more complex manipulation tasks. An exciting direction for future work would be to extend our results with multiple fingers equipped with touch sensors, with a joint predictive model that can predict the dynamics of object interaction with all of the fingers at once. Such setting could perform complex in-hand object repositioning, assembly, and other manipulation skills.

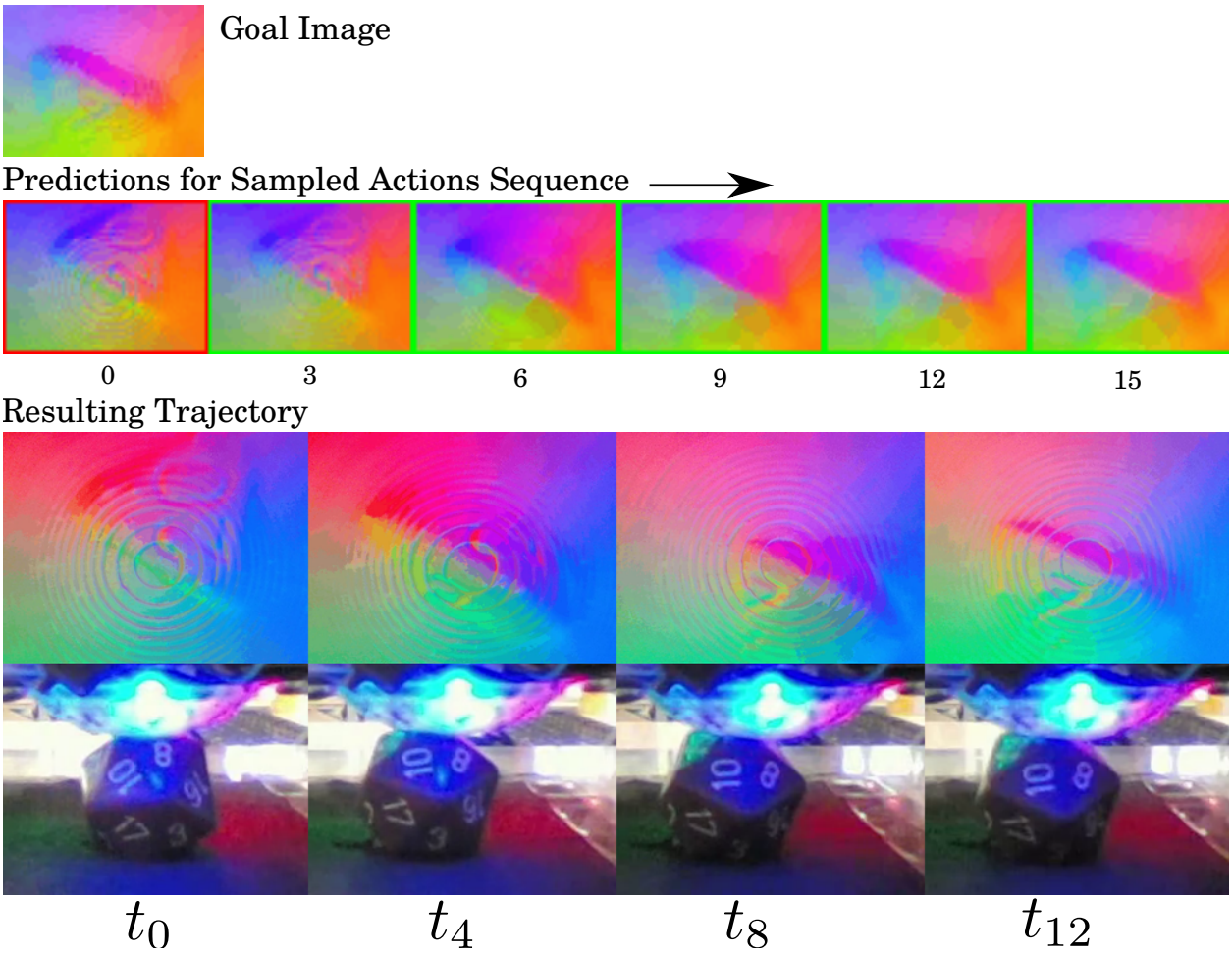


Figure 7.9: Example of successful execution of die rolling task. Starting from face 20 the goal is to reach face 8. The second row shows the video-predictions (at every 3rd time-step) for the best action sequence found at the first real-world time-step. The red margins indicate real context frames, green margins indicate predicted frames.

7.7 Acknowledgements

We would like to thank Chris Myers from the CITRIS Invention Lab at UC Berkeley for help with building the custom GelSight sensor and the 3 axis test-rig. The work in this chapter was performed in collaboration with Stephen Tian, Frederik Eckbert, Dinesh Jayaraman, Chelsea Finn and Sergey Levine and was presented at the International Conference on Robotics and Applications, 2019.

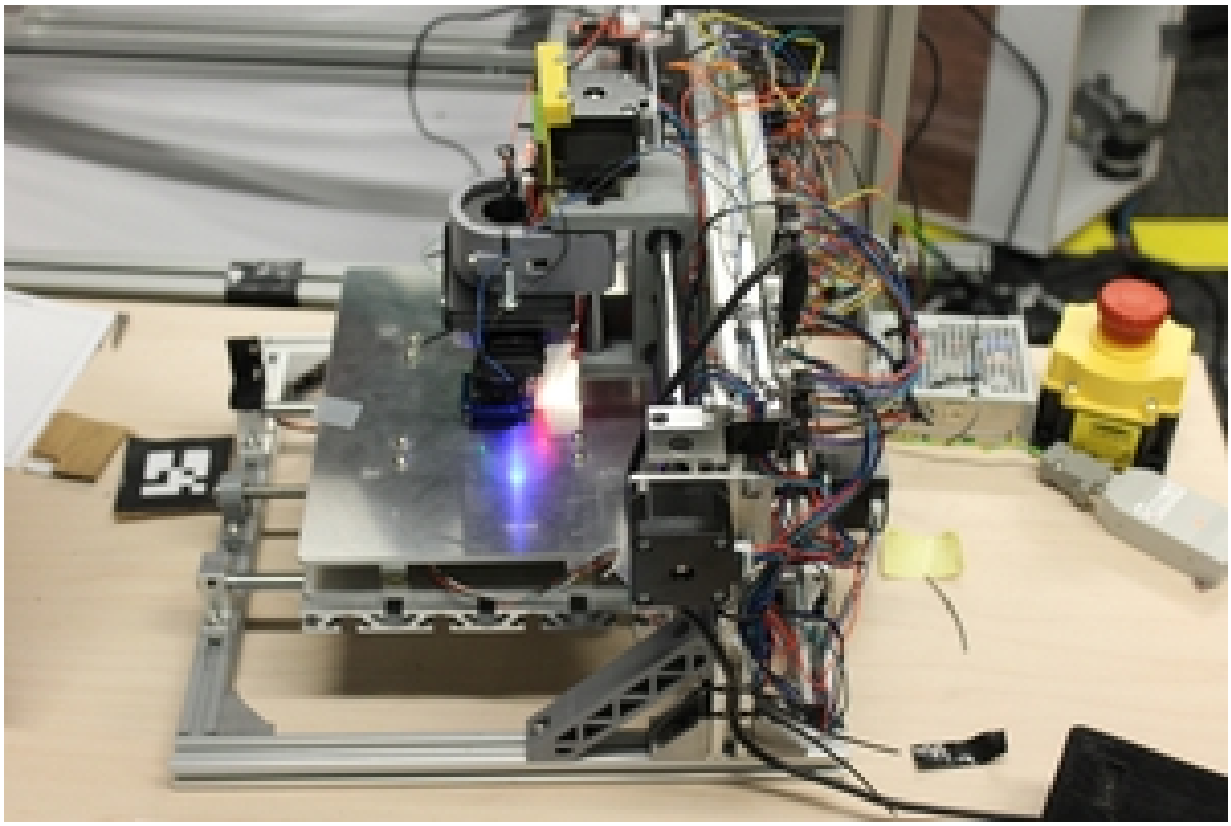


Figure 7.10: Hardware setup. Custom manufactured GelSight sensor mounted on a modified 3-axis CNC machine, which allows for linear translation along each of the three axes.

7.8 Appendix

Autonomous Data Collection

When performing autonomous data collection we either need to reset the environment to a well-defined set of starting conditions after each trajectory or the set of reachable states in the environment needs to be confined. In the case of the ball-rolling task we use slightly curved surface so that upon completion of a trajectory the ball automatically rolls back to a location close to the middle of the arena. For the analog-stick task a reset mechanism was provided by the springs embedded into the analog stick. In the die rolling task we used a thread fastened to the die wound onto a motor that resets the die to an approximately fixed starting pose at the beginning of each trial. For each trial, the sensor makes contact with the surface of the die and moves it in an arbitrary direction resulting in different die faces. At the end of each trial, the die is reset as described above.

To that end, we collected 7400 trajectories for the ball, around 3000 trajectories for the analog stick, and 4500 trajectories for the die experiment. Both during data collection and planning the actions are parameterized as changes in finger position of $\pm$ 6.0mm in the x, y, and z directions. Data is collected at 1.5Hz. At test time, tactile MPC runs at around 1 Hz. Both during data collection and planning we repeat actions for 3 time-steps, but we record images at every time-step, providing advantages for planning as explained in the paragraph *Implementation details*. We found that having a higher frequency for images than

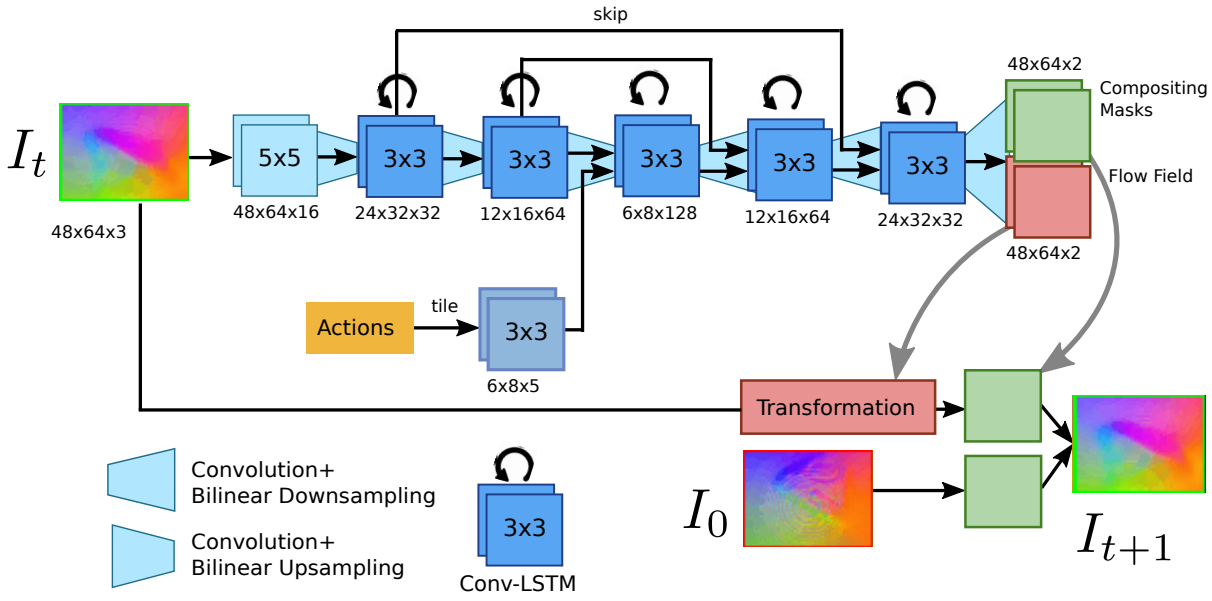


Figure 7.11: Video Prediction Architecture.

actions helps the model making more accurate predictions in environments with discontinuous dynamics.

Deep Recurrent Visual Dynamics Model

The video prediction model is implemented as a deep recurrent neural network. Future images are generated by applying transformations to previous images. A schematic is shown in Figure 7.11. More details on the video-prediction architecture can be found in [48] and [116]. Note that depending on the experiment during the first 3 time-steps of unrolling the RNN prediction model we feed the most recent ground truth observations, we call these images *context frames*.

Chapter 8

Investigating computational models to study the motor and barrel cortices in mice

8.1 Active sensation disrupts correlations in S1 and M1 networks in the mouse neocortex–a sensorimotor account

Animals function in a 3D world where repeatable, robust action drives their survival. Consequently, it is of great importance to understand sensorimotor representations and how sensory stimuli are represented and transformed into motor actions. Recent work [Matyas et al, 2010] has shown that there exists a very tight coupling between primary somatosensory (S1) and motor (M1) neurons in the mouse cortex but very little has been done to explain what sort of computations these populations of neurons might be performing. We present a preliminary analysis of new experimental data that was collected simultaneously from mouse S1 and M1. Network connectivity of active neurons was inferred from the coupling matrix of an Ising model fit to binary spiking data. We find that when mice actively palpate an object using multiple whiskers, S1 units become weakly coupled while M1 units appear to reorganize their couplings. These stimulus induced network decouplings may be carrying out the computations involved in sensorimotor transformations and warrant further study. We fit linear and non-linear models of the spiking data to decode whisker data. We find that a trained recurrent neural network model is able to decode gross whisker tracking features with R2 scores greater than 0.9. These models provide insights into how these populations of neurons may be building a model of the world through

8.2 Significance

The sense of touch is a very important faculty and it helps an organism understand various physical properties of the world as shown in Figure 8.1.

Organisms are not passive actors in their environment but are able to influence their surroundings and drive active changes to the stimulus they receive. The study of of feedback networks and sensorimotor couplings [171, 141] largely remains unexplored in the computational community. One obvious way to

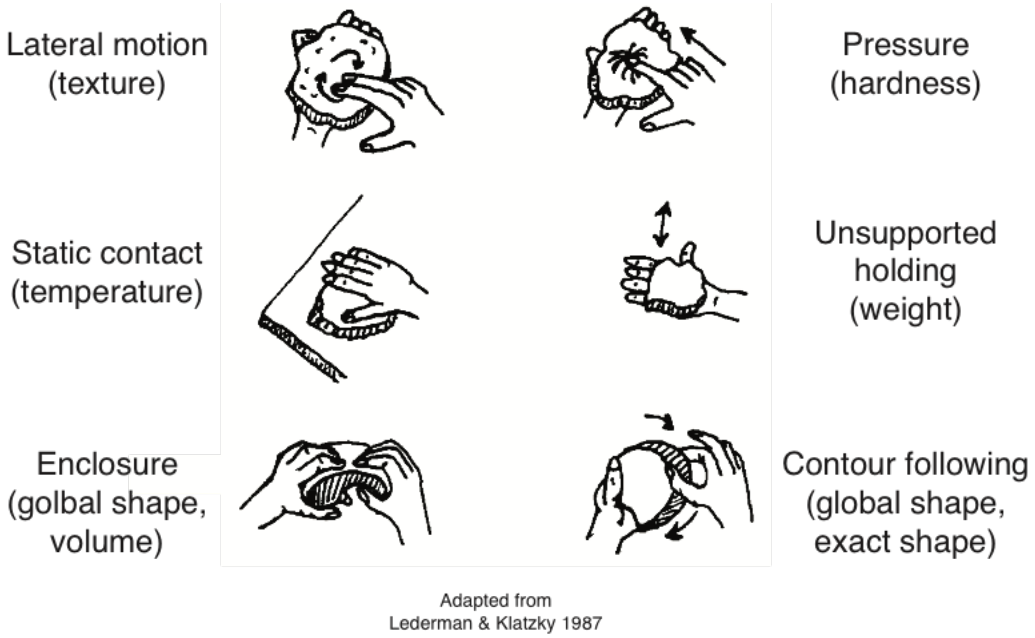


Figure 8.1: Human active sensing strategies from Lederman, 1987 [115]

address these emergent interactions is to model the sensory coupled motor systems simultaneously in a model biological system that relies heavily on sensorimotor interactions. S1 and M1 do not work in a simple feedforward way where S1 processes sensory information and passes it to M1 which then updates the motor program. Rather, both systems have been shown to not only have robust sensory responses but can induce and modulate whisker motion [131]. Since sensory and motor information are simultaneously represented in various cortical regions it is important to utilize models that will allow us to explore these representations in these regions while the animal is freely behaving.

Model Fitting

$$E(X, J; b, \lambda) = -X^T JX - b^T X - \lambda \|J\| \tag{8.1}$$

Ising models 8.1 [167] are traditionally fit through maximum-likelihood learning which requires Gibbs sampling for large models. Unfortunately, because Gibbs samplers updates each unit of the model sequentially, this process is slow. In our work, we apply Minimum Probability Flow (MPF) [178] method to fit Ising models to the data with a sparse L1 prior on the coupling matrix [77] which is approximately 100 times faster than fitting models with Gibbs sampling. 1950 samples for each neuron were used to fit the models in our study. Figure 1 shows the coupling matrices that were fit to the data.

8.3 Experimental Setup

Mice expressing channelrhodopsin (ChR2) in parvalbumin positive (PV) inhibitory interneuronwere were head fixed and placed on a circular treadmill as shown in Figure 8.2. While the mice ran freely a vertical stimulus bar was placed at various positions where the animal could actively touch the bar. The subsequent sensory activity was recorded using two 16 channel linear silicon electrodes placed in a single "barrel" column in S1 and in the vibrissae sensory region of M1. These two regions are known to have reciprocal anatomical connections [Matyas et al. 2010 [131]] making this the ideal place to study sensorimotor integration in the mouse. We then extracted the spike timings of 30 M1 units and 22 S1 units for the different stimulus

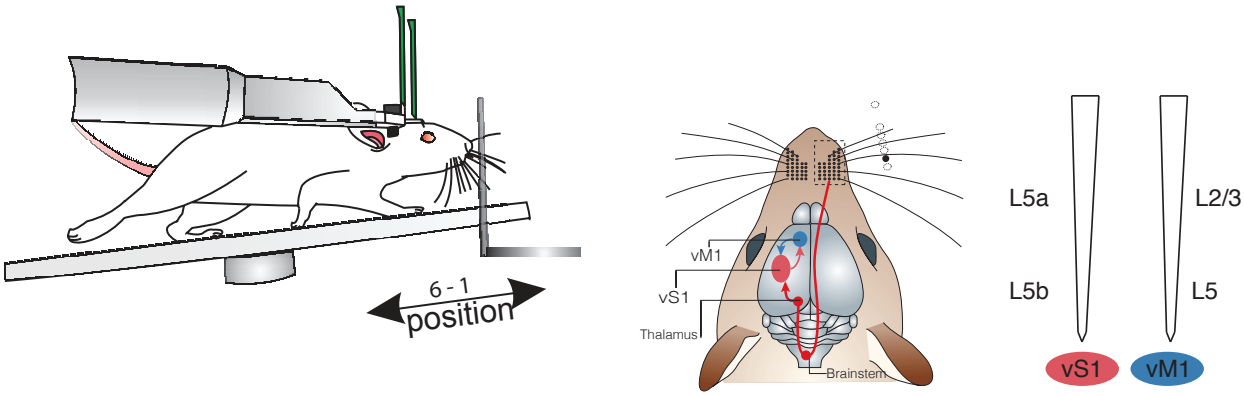


Figure 8.2: Experimental setup showing the mouse on a lazy susan with head fixed to the rig. Two insertions are made surgically to collect data from the cortex. The angle of the surface can be varied. A high speed camera is placed underneath to track whisker data. A controllable,movable bar is actuated to the desired location present stimuli

conditions. This setup allows us to specifically probe how sensory information is initially represented in somatosensory cortex and how it is transformed and represented in a region where this information is integrated to update motor output.

8.4 Results

Matyas et al. [131] showed that in the absence of motor activity whiskers may protract as seen in Figure 8.3. Further, it data collected show that there is specificity in the whiskers as seen in Figures 8.4,8.5. To better understand the population dynamics of these neurons we explored fitting Ising models. An important consideration to fit ising models is to choose the irght bin size. Figure 8.6 shows the various bin sizes and their model fits for free and active whisking. This was used to choose an optimal bin size for analysis. To better understand the model fits, we compared them against an independent binomial model and a Poisson model. These fits are shown in Figure 8.7.

We found that in the cases where whiskers contact a stimulus bar, the induced network in both S1 and M1 drastically change. Specifically, S1 units which have strong (magnitude) couplings during free whisking and an increased firing rate (3x as compared to free whisking) have weakened couplings between neurons during active palpation. In contrast, M1 units see a re-organization of their couplings and much less weakening of couplings as compared to S1 neurons, even when the firing rate increased (2x as compared to free whisking).These results are best described by the Ising model fits shown in Figure 8.8. When these populations of neurons were analyzed as one system, we see that there exist non-negligible couplings between the two subpopulations of neurons. Further, these also show similarly decreased couplings for active whisking.

Further, evidence from [131] and from the evoked firing rates of our data suggests that there exists a weak but non-negligible motor coupling to the whisking process. It will be interesting to determine how the pairwise interactions change amongst the units that contain whisker representations in S1 and whether we can predict which units will contain whisker information based on the changes in the pairwise interactions in future experiments. Whereas S1 exhibited a strong reduction in couplings, M1 showed only a modest

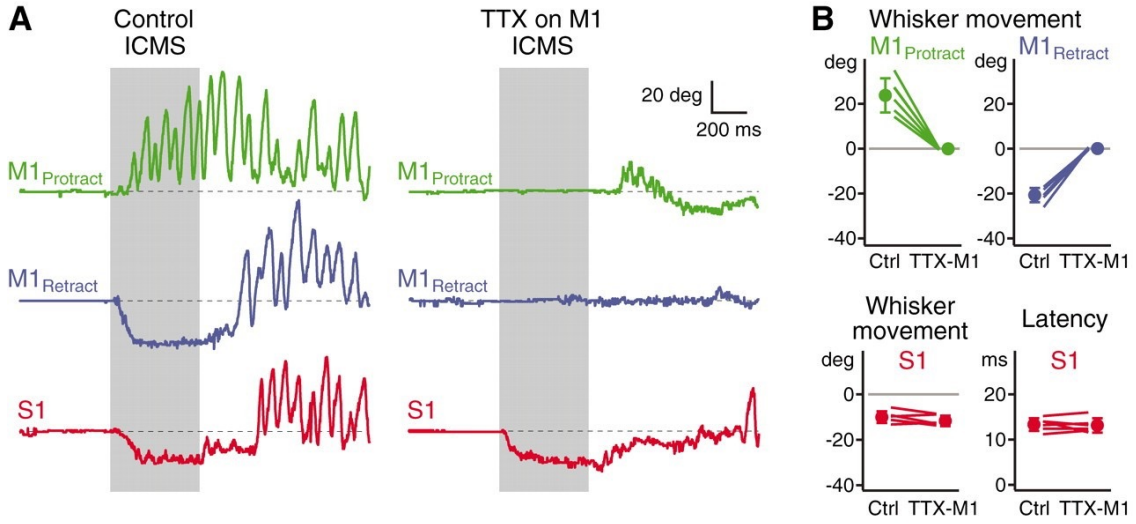


Figure 8.3: Whisker movement via vS1 with the absence of vM1 activity from [131]

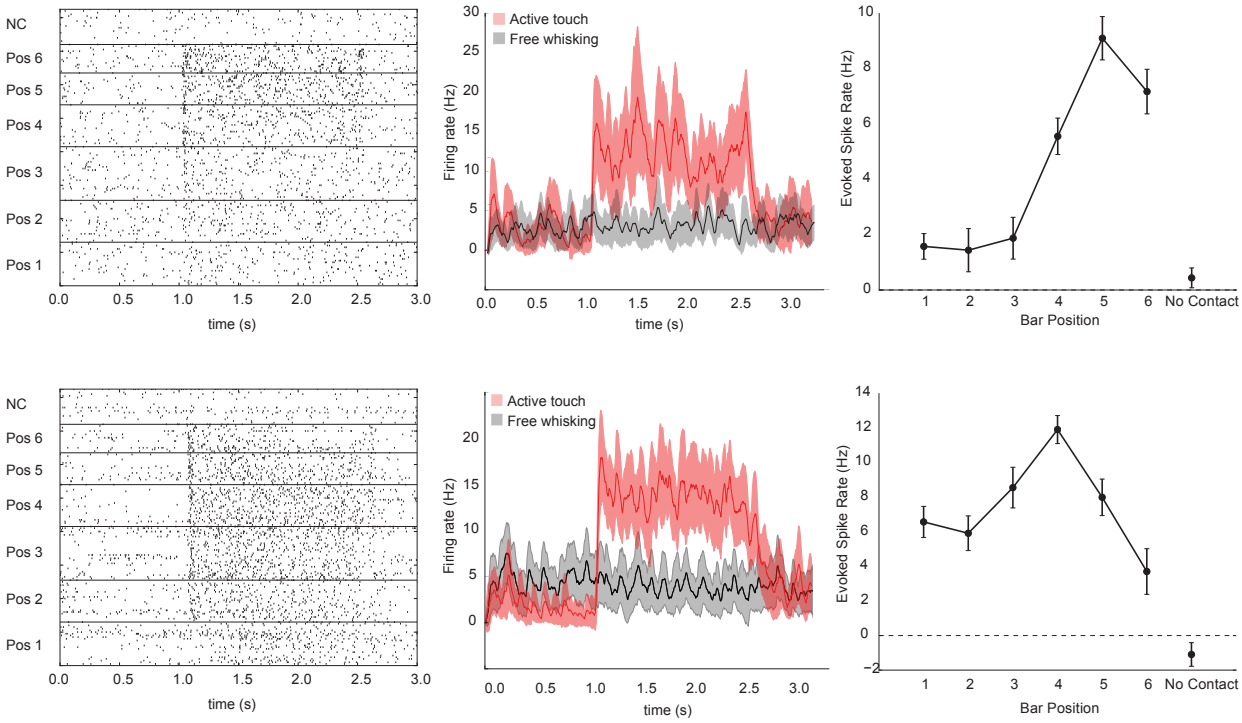


Figure 8.4: Example raster plots, Peri Stimulus Time Histograms (PSTH), spatial tuning curves for vM1 (top) and vS1(bottom)

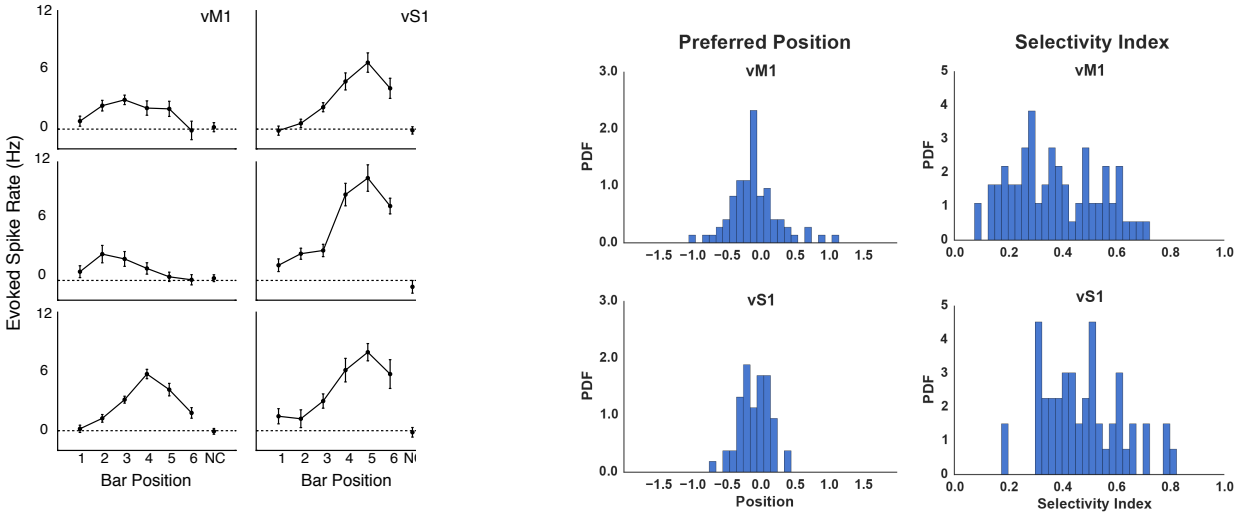


Figure 8.5: Preferred positions and Selectivity index

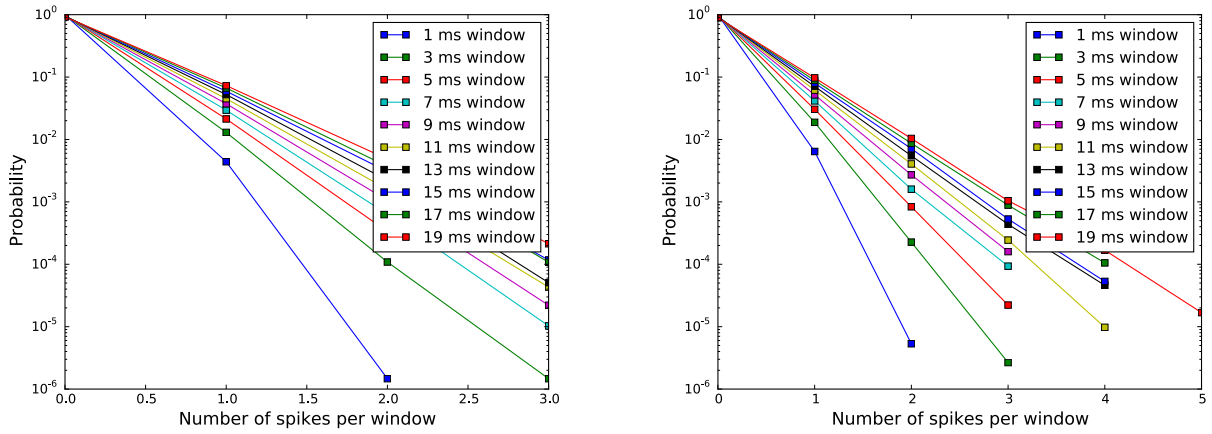


Figure 8.6: Choosing bin size (time) for fitting Ising models. (Left) Free whisking results are presented. (Right) Active whisking results are presented

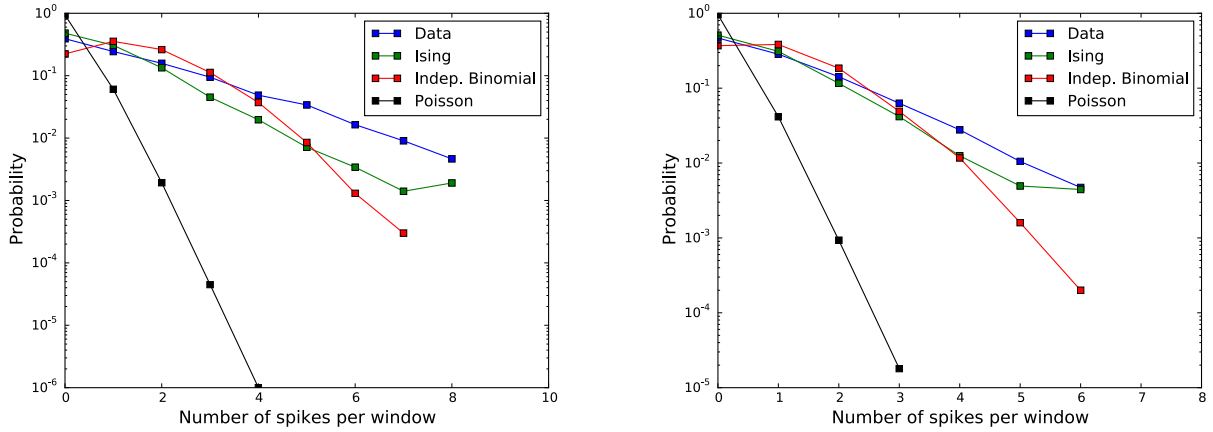


Figure 8.7: Comparing samples generated from three different models against the data distribution. We compare a Poisson, independent Binomial and Ising models. (Left) Free whisking results are presented. (Right) Active whisking results are presented

increase in decouplings. Additionally, it seems that units coupled during free whisking become coupled to different units during active sensation. Therefore, incoming sensory information appears to reorganize the interaction structure of the M1 sensory network. One concrete direction we wish to explore is building models that account for temporal delays in activation. Additionally, Figure 8.3 suggests jointly modeling the statistics of S1 and M1 could lead to insights into regions from which we are not collecting data from (e.g. the reticular nucleus, secondary somatosensory cortex S2) and the larger sensorimotor representations that mice use to explore 3D worlds.

8.5 Decoding whisking information in awake behaving mice from S1 and M1 neurons using neural networks

Recent work [131] has shown that there exists a tight coupling between primary somatosensory (S1) and motor (M1) neurons in the mouse cortex but very little has been done to explain the computational mechanism through which these populations of neurons might be building a model of the world. We present a preliminary analysis of new experimental data that was collected simultaneously from mouse vibrissae S1 and M1. We fit linear and non-linear models of the spiking data to decode whisker data. We find that a trained recurrent neural network model is able to decode gross whisker tracking features with R2 scores greater than 0.9. These models provide insights into how these populations of neurons may be building a model of the world through whisking.

Initial Findings

Interestingly, silencing either S1 or M1 had no effect on whisking behaviors. Our current hypothesis is that, in the absence of a behavioral task, cortex is not involved in controlling whisking behaviors. Rather, a central pattern generator in the brainstem has been shown to elicit rhythmic whisking [McElvain et al.

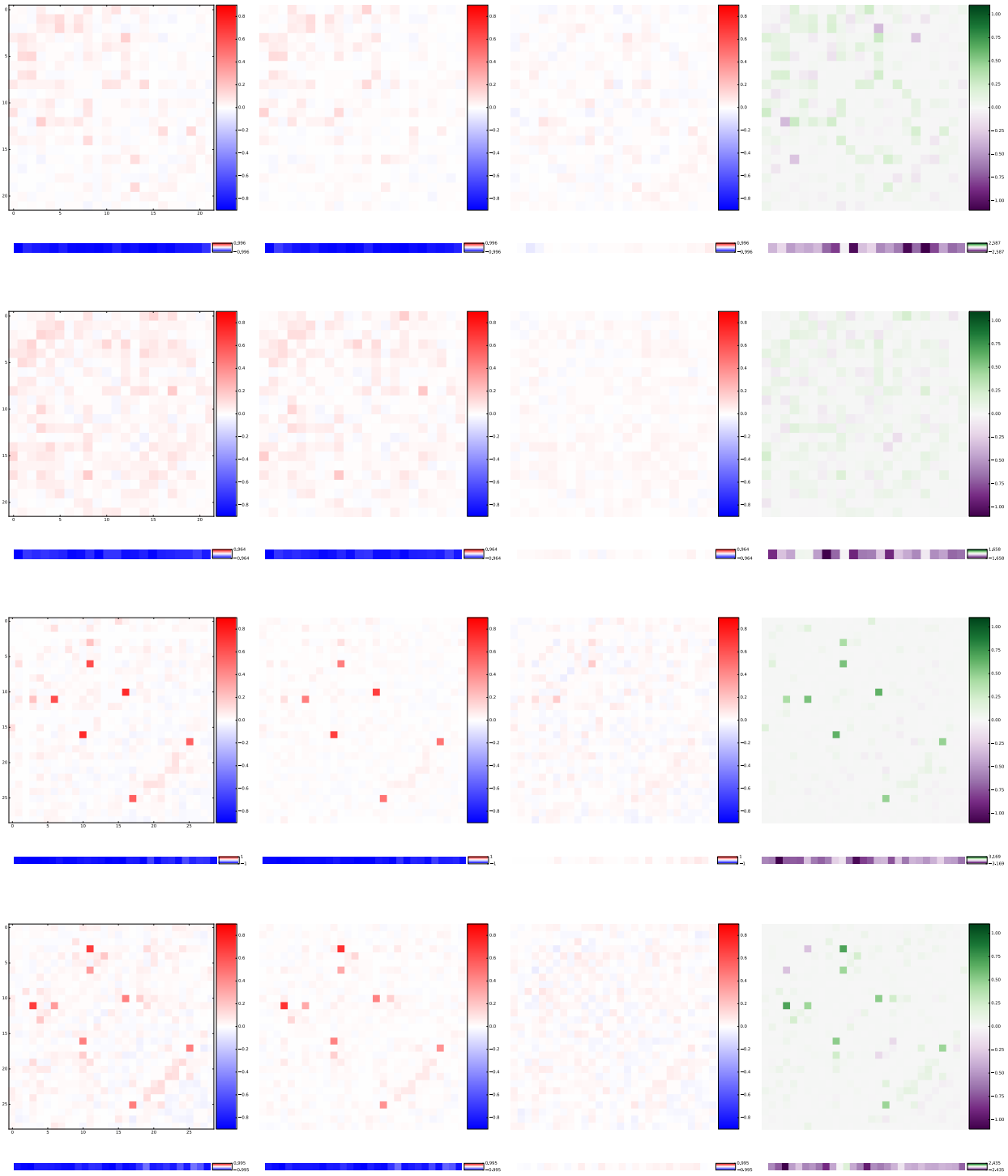


Figure 8.8: (Left to right) Data covariance, Model covariance, Difference between the two covariancs, and the Ising coupling matrices. (Top to bottom) Free whisking vS1 with no contact condition, Active touch vS1 where a strong contact is made, free whisking vM1 with no contact, active touch with vM1 where a strong contact is made

Model vs R^2	angle	set pt	ampl.	phase	vel	whisk	cond
LSTM	0.99	0.99	0.97	0.57	0.59	0.99	0.97
DNN	0.96	0.97	0.66	-0.00	-0.03	0.97	0.61
Wiener Filter	0.84	0.85	0.63	0.05	0.01	0.85	0.58
Wiener Cascade	0.92	0.92	0.64	0.05	0.01	0.92	0.59
XGBst	0.931	0.935	0.67	0.03	0.01	0.93	0.59

Table 8.1: Performance of linear filters, deep neural networks, and recurrent networks to predict various whisking features

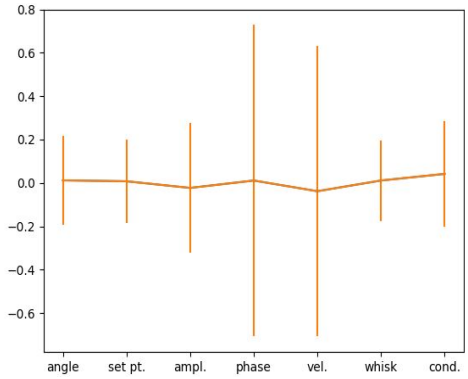


Figure 8.9: Comparision of performance of the LSTM decoding algorithm (vertical axis) on the various physical features such as whisking phase, velocity, angle, etc (horizontal axis)

2017]. Currently, we are training mice on a whisker-dependent operant task. This will allow us to assess whether the cortex is required for changing goal-directed whisking. Although whisking did not change, neural activity did. Silencing either S1 or M1 decreases sensory-evoked activity in the non-silenced region. Using a simple linear decoder we are able to predict stimulus position from the mean firing rates of all units with performance similar to the linear models described below. Silencing S1 typically reduces the performance of the decoder when trained and tested on M1 data. From this we can see that there is spatial information present in the mean rates of our units. To further understand how this, as well as whisker kinematics, is encoded in these two neural populations we decided to use different modeling techniques that can utilize both rate and temporal information. This will allow us to better understand what information is present in the two interconnected brain regions and determine how encodings change when one region is silenced.

Models,Results,and Discussion

Table 1 The Long Short Term Memory (LSTM) model outperforms other models. Summary of models and their R2 performances for each feature. We decoded the angle of the whiskers, set point (the sector swept out by the whisker pad), amplitude, phase, velocity, and whisk (a low pass filtered version of the angle signal). We also included the condition number as another feature. The models (from top to bottom) are an LSTM and a Deep Neural Network (with 500 units), a linear model (Wiener filter), a linear-nonlinear model (Wiener Cascade), and a boosted tree. Figure 1 shows the residual plot for each of the features for the LSTM model. The model struggles to predict velocity and whisking.

In our earlier work, we studied the population activity in the sensory (S1) and motor (M1) regions by fitting Ising models to spiking data. That work largely ignored the temporal structure and joint population coding effects, which are critical for modeling this data. Here, we contrasted various linear and non-linear methods for decoding spike trains. More details of the various methods can be found in Glaser et al. [65]. Spikes and whisker data were sorted and binned into 10ms bins. We then regressed this data onto whisking features. We used six time bins before and after the current time bin as inputs. We had approximately 32,000 samples for training. A further 7000 samples each were kept aside for validation and testing to prevent overfitting or biasing the model. We found that a recurrent neural network (LSTM) worked the best and was able to decode both phase and velocity by leveraging temporal information, which other models failed to do. Interestingly, we were able to decode whisking information well even with S1 or M1 silencing, suggesting that the information is either redundantly represented or that it originates in some other brain region.

To better understand this circuit, in future work we hope to fit GLMs convolved to spatio-temporal bases. This may provide some insights into the nature of computation. Further, exploring ideas such as visualization of LSTMs may provide insights as well. It is an exciting time to explore this area of representations with the advancement of both recording technology and statistical tools.

8.6 Acknowledgements

The work in this chapter was performed in collaboration with Greg Telian, Jesse Livezey, Ryan Zarcone with advise from Mike Deweese and Hillel Adesnik. A portion of this work was presented at the annual Computational and Systems Neuroscience conference at Salt Lake City, Utah, Feb 2016.

Chapter 9

Discussion

We, as a community, have made great strides in building systems that perform more complicated sensorimotor tasks. That said, what biological agents do still far outperforms what we can currently perform.

For example jumping spiders with only ≈ 35000 neurons can perform complex tracking, navigation and hunting (manipulation) [47, 111, 36] in an extremely robust fashion that far outperforms any modern day Artificial Intelligence system. This is a truly remarkable feat.

What is particularly impressive is that these actions are performed with noisy sensors and noisy actuators. It then follows that there remain many undiscovered principles that aid in this behavior and warrant further study.

Understanding the relationship between populations of neurons in barrel and motor cortices

From a neuroscience perspective understanding the nature of representations and computations performed by the sensorimotor cortex could potentially help us understand principles of the cortex. In our work so far we attempted to demystify the relationship between the barrel and motor cortex but we have awhile to go. One obvious direction to pursue would be fitting generalized linear models to the population of neurons and understanding the spatio-temporal structure of the data.

Further, to ascertain the role of the cortex looking at experiments that involve learning might prove useful. We know that for a lot of motor activity older structures such as the central pattern generator or other lower brain structures are important but we know relatively little on the relationship between the cortices and these structures.

Implicit 3D

Animals and machines function in 3D worlds and consequently must have an internal representation of this world. This representation is built from multiple views[79], having implicit priors [185], and more .

An important assumption in most 3D representations is the need for euclidean depth information as ground truth data during the learning of the model. This is a bit of a strong assumption as far as biology is concerned because such information is not available to biological agents.

What animals do have is the availability of other sensory modalities and the ability to test and better its hypothesis by acting on the world. One might even call this implicit 3D representation (pertaining to shape [15]).

One could imagine then to learn a 3D shape model without any ground truth euclidean depth data an agent could use its sense of touch. For example, in the figure 9.1 we see a shape emerge from just visual inputs, a dexterous hand with touch sensors and probing the object. After a certain amount of exploration, the picture on the right emerges which is a coarse shape representation of the object. Note, the depth information here is really a function of the motor program. In other words, how much torque did I have to apply to this joint to reach the object.

What is unique about this representation is that the agent has no access to acutal depth information and would work just as well with noisy sensors and actuators and the shape learnt is also a direct function of the actuator employed. One could argue that this is somewhat akin to the representation learnt by Zipser and Andersen [203]. We know that there are multiple representations of the object in the cortex (eye centric, head centric and so forth). The idea we present here is similar in that we could have object representation in different actuator spaces and the transformations to go between the various spaces.

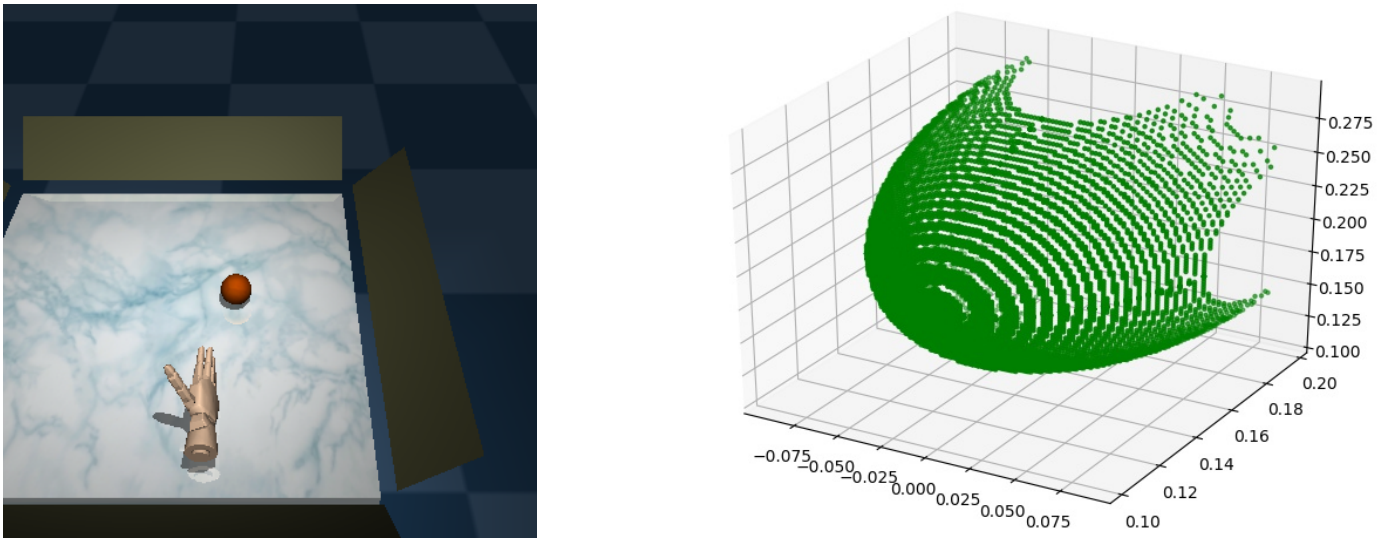


Figure 9.1: (Left) An anthromorphic dextrous hand with touch sensors along with an object (sphere) that we wish to explore. (Right) an implict 3D shape representation that is learnt through probing the object

Advancement of hardware in tactile sensing

Recent progress in image classification problems [105] was in large part inspired by the availability of new hardware that could perform certain type of operations (matrix multiplications) extremely fast. Similarly, while most roboticisits acknowledge the importance of tactile sensing it is the availability of hardware and the possibility of including it in the learning model that has stalled the advance of tactile sensing in learning based robotics.

The field of tactile sensors is an active interdisciplinary research field drawing from multiple domains such as mechanical engineering, material sciences, optics and more. For a recent survey of various sensing technologies one could peruse the following paper [204]. One of the key challenges for robust tactile sensing is wear and tear of the hardware along with their sensitivity. Different tactile sensors are capable of measuring

different types of changes such as pressure, shear, temperature and more. In contrast human tactile sensors are capable of measuring many different physical properties simultaneously.

More recently, there has been some exciting work in using foldable electronics to create a scalable haptic glove [182]. Works along these lines might help bridge the gap and open up the possibility of doing interesting tactile research both with humans and robots.

Learning motor representations through unsupervised learning

An open problem in the fields of robotics, controls and motor representations is how do we learn to actuate high dimensional, dextrous actuators. Traditional control methods do not scale as well in high dimensions but pure reinforcement learning methods are also challenging in terms of their sample complexity and somewhat brittle nature of the representation. For example, a representation learnt to lift a cup does generalize for other objects in humans but not so with our current representations.

I propose that the solution might lie in the exploration of unsupervised learning and supervised learning methods. One can imagine learning unsupervised motor primitives through goal or motor babbling. These primitives can be thought of as spatiotemporal primitives actuating a set of end effectors in a specific configuration for a period of time. One can then apply reinforcement learning methods to learn policies on these discrete primitives for any specified objective.

The benefit of such a representation is that it would easily generalize to new tasks and lends itself to elegant hierarchical motor learning in addition to maintaining a low sample complexity for learning.

Fitting probabilistic models with MCMC sampling

In this work we have presented efficient methods that can help sample from distributions but have not focused on their applications. As we discussed in the introductory chapter presently MCMC methods are the only way to full approximate the distributions. Further, the class of methods we have put forward scale well to continuous variables and high dimensions.

In the age of deep learning it would be interesting to explore deep, hierarchical models trained with sampling as opposed to maximum a posteriori estimates to see if they generalize better. One could imagine training two or three layer networks to begin. In fact, an obvious model could be training a product of experts model [146]

The laplacian group sparse coding model [59] is another obvious choice. In their work, the authors use a variational approximation to what is in effect a two layer group sparse coding model using laplacian priors. We could potentially train the network without the variational approximation with sampling methods and study the robustness of representations.

Exploring, MCMC sampling using analog devices could also be a very promising way to explore real time inference for complex probablistic models.

It would also be very interesting to combinine energy based functions for policies with HMC sampling. For example, in [75] one could imagine having an HMC type sampler for the exploration phase of the policy learning which might lead to exploring novel states faster perhaps.

Appendix A

Learning non-local features using compressed sampling

We explore whether a recent dimensionality reduction technique called adaptive compressed sampling (ACS) can discover non-local features useful for classification. The ACS scheme involves first reducing the ambient data dimension by multiplying with a fixed random compression matrix (as in compressed sensing), and then performing unsupervised sparse dictionary learning. Features in the original space are obtained by cross-correlating latent representation variables with the original data. Our results demonstrate that classification performance on MNIST using latent variables inferred after learning remains high for a large range of compression, supporting recent theoretical and experimental findings. Further, we show that the learned features are more class-specific than local sparse features, and thus help characterize shape properties of the different classes.

A.1 Introduction

Olshausen and Field introduced sparse dictionary learning (or sparse coding) [143] as a method of unsupervised feature learning for image patches. Since its inception, sparse coding has been applied to reveal latent structure in diverse types of sensory data, including natural images [143, 156], natural sounds [175, 27], and even in the paintings of Pieter Bruegel [88]. Recently, it has also been discovered that sparse codes inferred from data are good features for certain discrimination tasks, including many that are vision related [195, 17, 37, 157]. Ideally, sparse coding can aid classification and also make the class structure transparent by extracting class-specific features.

Since sparse dictionary learning algorithms become computationally expensive as the data dimension grows, images are commonly broken down into small patches and sparse coding is employed to extract local features within these patches, e.g., [17, 37, 109]. The entire image is then represented as a concatenation of the local sparse codes. Another approach is to use large-scale computing (e.g. 16,000 cores) with deep autoencoding networks (e.g. 9 layers) to learn such features [112]. Such computations can provide substantial boosts in classification performance. For instance [112] achieves a 70% increase in performance versus state-of-the-art on a challenging dataset (15.8% on 20,000 object categories from ImageNet).

Here, we propose to use principles of compressed sensing (CS) as a novel alternative to discover features in high-dimensional data. The field of CS studies the conditions under which latent or original sparse codes can be reconstructed from compressed data given a known dictionary [25, 45]. Our approach is to combine the dimension reduction properties of compressed sensing with unsupervised dictionary learning (sparse coding) to compute latent representations from compressed data. If these latent representations are useful

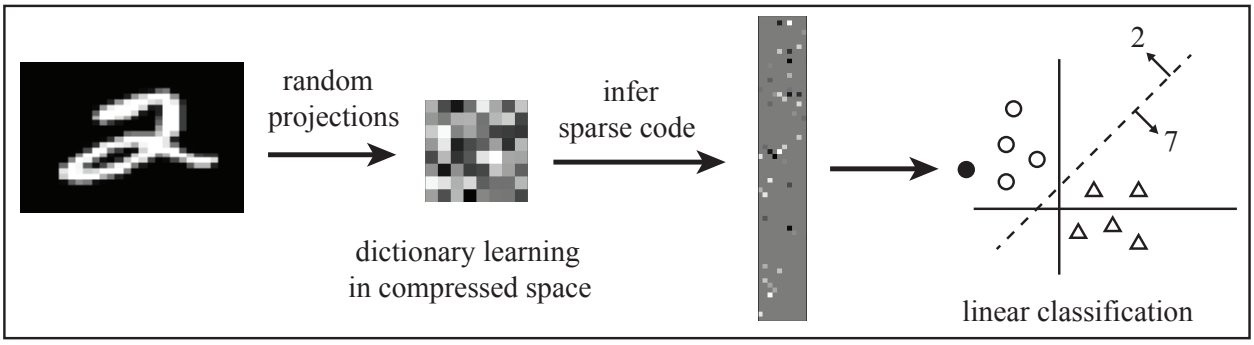


Figure A.1: Using sparse codes in a projected space for hyperplane discrimination of MNIST digits.

for tasks on original data, such as classification, then dimensionality reduction has succeeded. A feature of the method is that it does not require access to original dictionaries or compression matrices, which is important for some datasets.

In the literature, this two-step paradigm has been called adaptive compressed sensing (ACS). The theory of ACS guarantees that original sparse codes of certain kinds of data can be recovered (up to a fixed permutation and scaling) by dictionary learning from linearly compressed data [4, 90, 82]. Thus, rather than dividing high-dimensional data into chunks (image patches), employing large-scale computing, or multi-layered networks, we utilize compressed sensing to reduce the data dimension so that dictionary learning is easier to converge. Note that this form of dimensionality reduction does not confine the sparse features to be local. Although the learning in ACS does not automatically produce features of the uncompressed data, simple reverse correlation can be employed to construct them, once learning has converged (see Figure A.2).

As an important proof of concept, we apply ACS to classification of the MNIST data set [113] on which conventional local coding of image patches has been demonstrated to be efficient [109]. Our first main result is that even high ($10\times$) compression rates enable classification results that are commensurate with those achievable by sparse coding on original data. Our second main result is that many of the resulting non-local features reveal characteristic shape properties of members from specific classes. Our findings suggest that the extraction of non-local features in many types of high dimensional data sets could be made tractable by applying ACS.

A.2 Background

Sparse signals. Several studies have shown that natural signals falling onto sensory organs have a higher-order structure that can be well-captured by sparse representations in a basis learned using unsupervised dictionary learning. See [162, 143, 156] for visual input and [11, 175, 27] for auditory. We define such special classes of signals formally.

Definition 1: An ensemble of signals X within $\mathbb{R}^n$ has *sparse underlying structure* if there is a dictionary $A \in \mathbb{R}^{n\times p}$ so that any point $\mathbf{x} \in \mathbb{R}^n$ drawn from X can be expressed as $\mathbf{x} = A\mathbf{a}$ for a sparse vector $\mathbf{a} \in \mathbb{R}^p$.

We consider an *ensemble* of random vectors with sparse underlying structure that arises from a probability distribution, although for data sets in vision (e.g. natural image patches) we cannot guarantee this to be the case.

Compressed sensing with a fixed basis. Compressed sensing or compressive sampling (CS) [25, 45] is a method for representing data with sparse structure using fewer samples than required by the Nyquist-

Shannon theorem. In the formulation of [188], a signal $\mathbf{x} \in \mathbb{R}^n$ is assumed to be *k-sparse* in an $n \times p$ dictionary matrix Ψ; that is, $\mathbf{x} = \Psi\mathbf{a}$ for some vector $\mathbf{a} \in \mathbb{R}^p$ with at most k nonzero entries. Next, $\mathbf{x}$ is subsampled using an $m \times n$ compressive matrix Φ to give noisy measurements $\mathbf{y} = \Phi\mathbf{x} + \mathbf{w}$ with $m \ll n$ and independent noise $\mathbf{w} \sim \mathcal{N}(0, \sigma^2 I_{m\times m})$. To recover the original signal, the following "Lasso" convex optimization problem is solved:

$$\widehat{\mathbf{b}}(\mathbf{y}) := \arg\min_{\mathbf{b}} \left\{ \frac{1}{2n} ||\mathbf{y} - \Phi\Psi\mathbf{b}||_2^2 + \lambda|\mathbf{b}|_1 \right\}, \tag{A.1}$$

and then $\widehat{\mathbf{x}} := \Psi\widehat{\mathbf{b}}$ is set to be the approximate recovery of $\mathbf{x}$. Remarkably, as can be shown using [188], the preceding algorithm determines a unique $\widehat{\mathbf{b}}$ and is guaranteed to be exact within the noise range:

$$||\mathbf{x} - \widehat{\mathbf{x}}||_2 = O(\sigma) \tag{A.2}$$

with high probability (exponential in m/k) as long as the $m \times p$ matrix

$$A = \Phi\Psi \tag{A.3}$$

satisfies mild incoherence hypotheses, and the sparsity is on the order $k = O(m/\log p)$.

Typically, the matrix Ψ is $p \times p$ orthogonal, and the incoherence conditions reduce to deterministic constraints on Φ only. Although in general it is very difficult to decide whether a given Φ satisfies these conditions, it is known that many random ensembles, such as i.i.d. $\Phi_{ij} \sim \mathcal{N}(0, 1/m)$, satisfy them with high probability [9]. In particular, compression ratios on the order $(k \log p)/p$ are achievable for k-sparse signals using a random Φ chosen this way.

Sparse dictionary learning by sparse coding. For some natural signals there are well-known bases (e.g. Gabor wavelets, the DCT) in which those signals have a sparse or nearly sparse underlying structure. However, an arbitrary class of signals can be sparse in unknown bases, some of which give better encodings than others. Sparse coding methods [143, 156] learn dictionaries for datasets by minimizing the empirical mean of an energy function that combines ℓ_2 reconstruction error with a sparseness penalty ($\lambda > 0$) on the encoding:

$$E(\mathbf{x}, \mathbf{a}, \Psi) = ||\mathbf{x} - \Psi\mathbf{a}||_2^2 + \lambda S(\mathbf{a}). \tag{A.4}$$

It is common to choose $S(\mathbf{a})$ to be the ℓ_1 penalty $S(\mathbf{a}) = |\mathbf{a}|_1 = |a_1| + \cdots + |a_p|$. Fixing Ψ and $\mathbf{x}$ and minimizing (A.4) with respect to $\mathbf{a}$ produces a vector $\widehat{\mathbf{a}}$ that approximates a sparse encoding for $\mathbf{x}$.[1] For a fixed set of signals $\mathbf{x}$ and encodings $\mathbf{a}$, minimizing the mean value of (A.4) with respect to Ψ and renormalizing columns produces an improved sparse dictionary. Alternating optimization steps of this form, one can learn a dictionary that is tuned to the statistics of the class of signals studied. Sparse coding on natural stimuli has been shown to learn basis vectors that resemble the receptive fields of neurons in early sensory areas [143, 156, 27]. Note that once an (incoherent) sparsity-inducing dictionary Ψ is learned, inferring sparse vectors $\widehat{\mathbf{a}}$ from signals $\mathbf{x}$ is an instance of the Lasso.

A.3 Adaptive Compressed Sampling

Adaptive compressed sampling is a technique motivated by bottleneck communication between distant neural systems in the brain [38, 90]. The idea is to randomly project high dimensional (but sparse in some basis) data into a significantly smaller space so that the information can travel through a fiber (dimension) bottleneck. The receiver region then performs sparse dictionary learning on the compressed data to produce a good dictionary of compressed data. Surprisingly, experimental [38, 90] and theoretical results, e.g., [4, 181, 82], demonstrate that the sparse codes representing the compressed data in the learned dictionary are the same

[1] As a convention here, $\mathbf{a}$ vs. $\mathbf{b}$ denotes a sparse representation inferred from full vs. compressed signals.

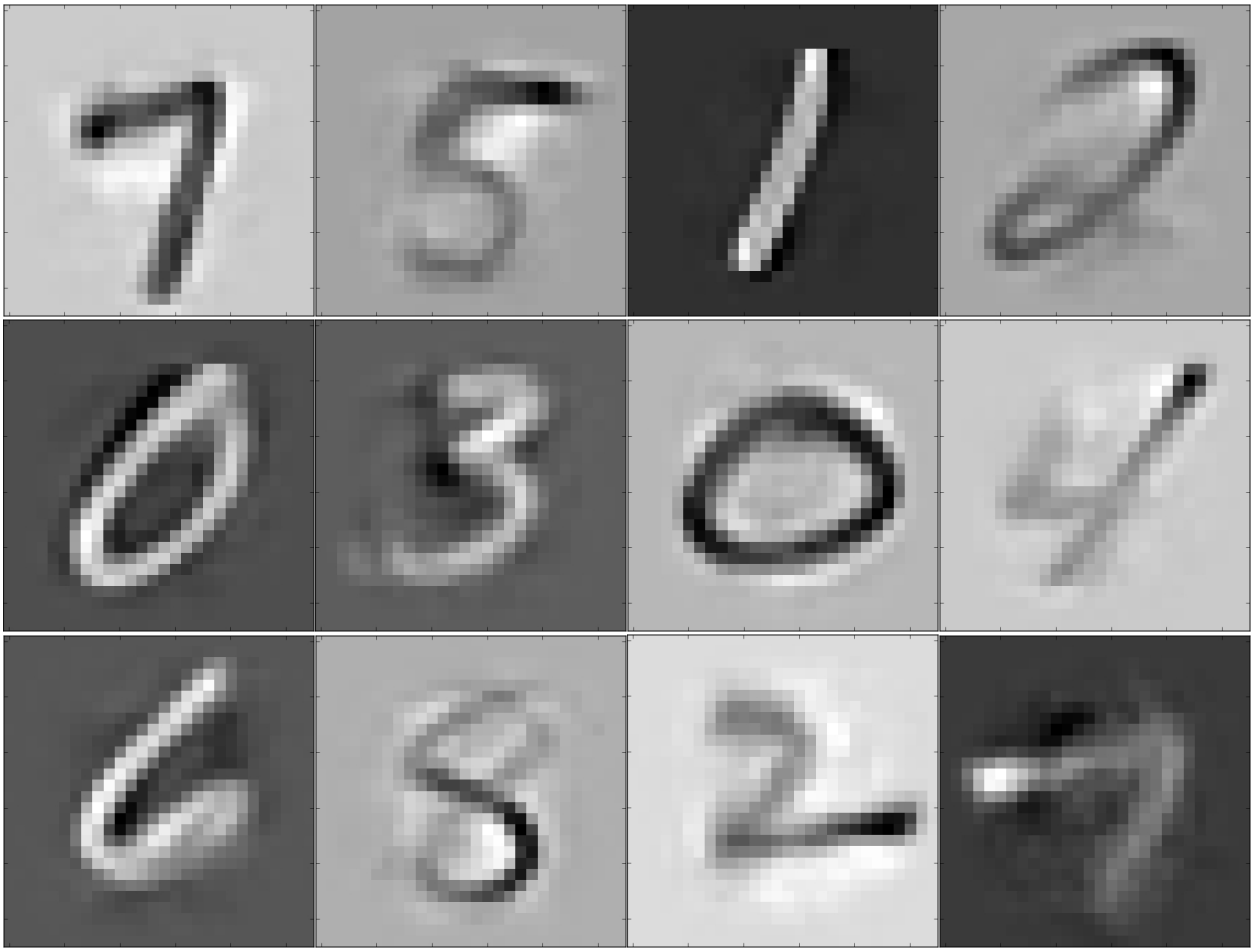


Figure A.2: **Non-Local learned features (STAs)**: We display some columns from a reconstruction matrix (A.7) formed from cross-correlating original digit images with coefficients inferred from a sparse coding model trained in a compressed space (original 28×28 images were compressed $10\times$ and then sparse coding trained with a $4\times$ overcomplete learned dictionary). It is surprising that by cross-correlating sparse codes learned in a compressed space with original 28×28 images we are able to uncover digits as the most salient features for a given coefficient in the code $\mathbf{b}$.

as the original sparse codes representing the data (up to a fixed permutation and scaling). We illustrate a small ACS simulation in the two panels below.

The ACS objective function on a compressed vector

$$\mathbf{y} = A\mathbf{a} \tag{A.5}$$

is defined as:

$$E(\mathbf{y}, \mathbf{b}, B) = ||\mathbf{y} - B\mathbf{b}||_2^2 + \lambda S(\mathbf{b}). \tag{A.6}$$

Here $B \in \mathbb{R}^{m \times p}$ is a dictionary to be learned over the compressed versions $\mathbf{y} \in \mathbb{R}^m$ of the data $\mathbf{x}$ (having underlying sparse cause $\mathbf{a}$). Iterated minimization of the empirical mean of this function first with respect to $\mathbf{b}$ and then with respect to B will produce a dictionary B for the compressed space and sparse representations $\widehat{\mathbf{b}}$ of the $\mathbf{y}$ [38, 90].

It has been demonstrated under some mild conditions that if the above procedure converges a dictionary that accurately represents the data in the compressed space, then there is a $p \times p$ permutation matrix P and an invertible diagonal matrix $D \in \mathbb{R}^{p \times p}$ such that $A = BPD$, and that for each sparse vector $\mathbf{a}$, the inferred $\mathbf{b}$ satisfies:

$$\mathbf{b} = PD\mathbf{a}.$$

In other words, sparse vectors that are compressed can be recovered exactly up to a fixed natural transformation (permutation and scaling) using sparse dictionary learning. However, as has been shown in [66], it is an ill-posed problem to compute basis functions Ψ of the uncompressed data directly from the learned matrix A. Since in this application of ACS there is also direct access to the uncompressed data, one can compute instead a *reconstruction matrix RM* [90] that minimizes the empirical mean of the reconstruction error and thus approximates Ψ by the closed form solution

$$RM = C_{sr}C_{rr}^{-1}, \tag{A.7}$$

with C_{sr} the data-response cross-correlation (or "reverse-correlation") matrix and C_{rr} the response autocorrelation matrix. It is common to call RM or C_{sr} a "spike-triggered average" (STA), as they do in experimental neuroscience.

See [57] for a discussion of applications of compressed sensing to neuroscience, more generally.

A.4 Experiments

To explore the ability of ACS to extract efficient and non-local classification features, we employ the method on the MNIST dataset [113]. The MNIST dataset is commonly used to test both feature spaces and classifiers in the computer vision and machine learning community. For example, it has been shown that dictionary learning on 13×13 patches yields efficient classification features [109] for MNIST. In this paper we explore how ACS can recover structure from this dataset even under significant random linear compression. In the following, we describe the specifics of our simulation experiments.

Experimental Pipeline

We took each of the 60000 training samples in the MNIST dataset and projected them onto a set of random basis ϕ. The size of the basis determined the compression of the input. We then applied a sparse dictionary learning algorithm to learn the basis on the random projected basis. We employed the SPAMS toolbox to do the lasso L1 inference of codes. For learning the basis, we employed a stochastic gradient descent with a batch size of 100. We employed a line search to compute the optimal step length to apply for the update rule. We trained all models for exactly 10000 iterations and noted no obvious change in reconstruction error after that. We set an expected sparsity of 2.5% for the inferred codes.

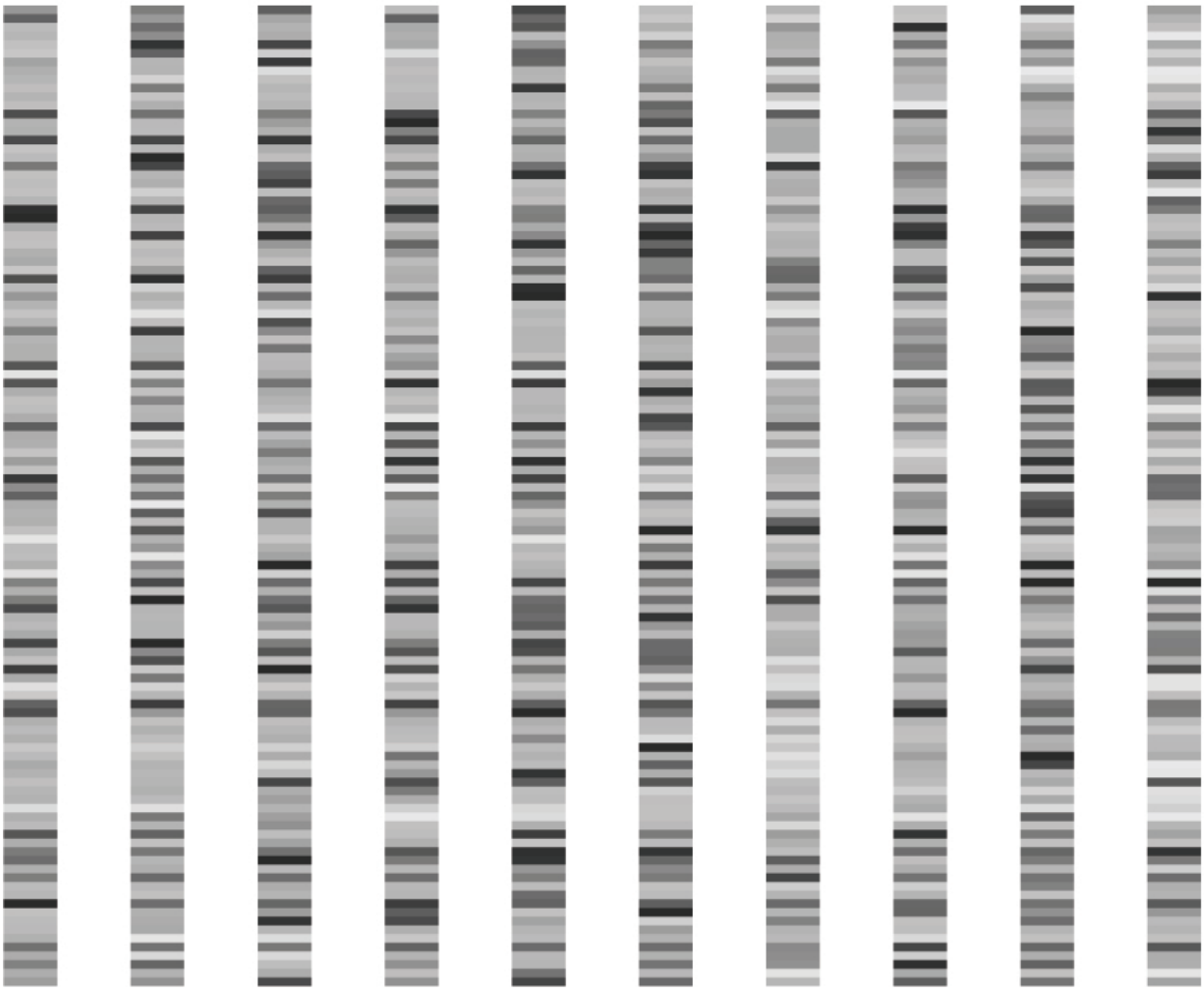


Figure A.3: Sample dictionary (88-dimensional) columns from training sparse coding on 9× compressed MNIST digits. The learned dictionary features in the compressed space appear to be random patterns. However, the codes this dictionary produces contain discriminative information (see Figure A.5 for classification performance using them and Figures A.2 and A.6 for non-local features they code) about the original uncompressed 28×28 (784-dimensional) input patches.

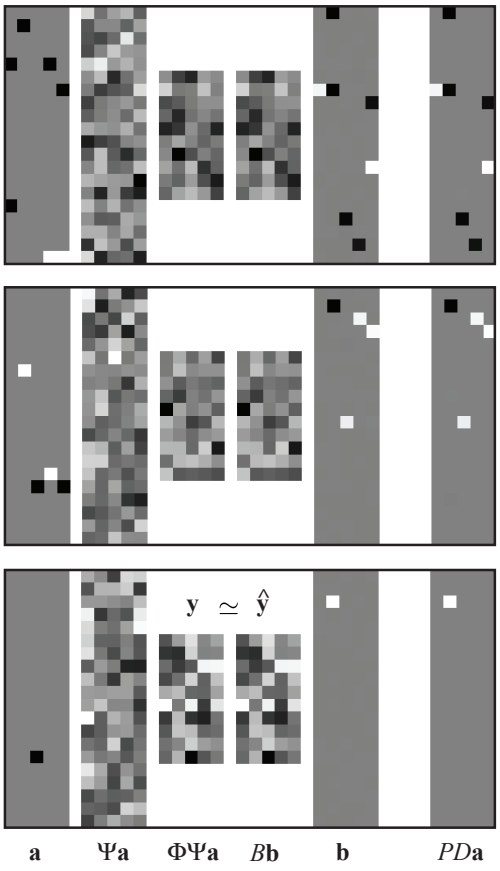


Figure A.4: **Dimensionality reduction with ACS**. Simulation testing the ability of ACS to recover sparse signals from compressed data. We first picked a random (normal) dictionary matrix $\Psi \in \mathbb{R}^{100\times100}$ and a random compression matrix $\Phi \in \mathbb{R}^{50\times100}$. We then created a training dataset of random vectors $\mathbf{a} \in \mathbb{R}^{100}$ with sparsities $k = 1, \ldots, 6$ (the number of nonzero entries of $\mathbf{a}$), and learned a dictionary $B \in \mathbb{R}^{50\times100}$ using sparse coding over compressed versions $\mathbf{y} = \Phi\Psi\mathbf{a} \in \mathbb{R}^{50}$ of the data $\mathbf{x} = \Psi\mathbf{a} \in \mathbb{R}^{100}$. As depicted in three representative examples, a network trained on the compressed data can fully recover the original sparse signals ($k \leq 7$) up to a fixed permutation and scaling. The 1st rectangle on the left represents a sparse 100-dimensional $\mathbf{a} \in \mathbb{R}^{100}$. White squares represent the most positive values for coordinates and black squares the most negative. The 2nd rectangle from the left is the sparse vector $\mathbf{a}$ represented in the dictionary $\Psi \in \mathbb{R}^{100\times100}$. For instance, the bottom-most example has only 1 coordinate in $\mathbf{a}$ nonzero and it is negative, so the data $\mathbf{x} = \Psi\mathbf{a}$ is simply the scaled column of Ψ corresponding to the nonzero entry in $\mathbf{a}$. Next, the signal is compressed using the compression matrix Φ to give $\mathbf{y} = \Phi\Psi\mathbf{a}$. The learned dictionary $B \in \mathbb{R}^{50\times100}$ now codes for $\mathbf{y}$ with a sparse vector $\mathbf{b}$, giving reconstruction $\hat{\mathbf{y}} = B\mathbf{a}$. To verify ACS working, we find a (fixed) permutation P and diagonal D with $\mathbf{b} = PD\mathbf{a}$ (right-most rectangle).

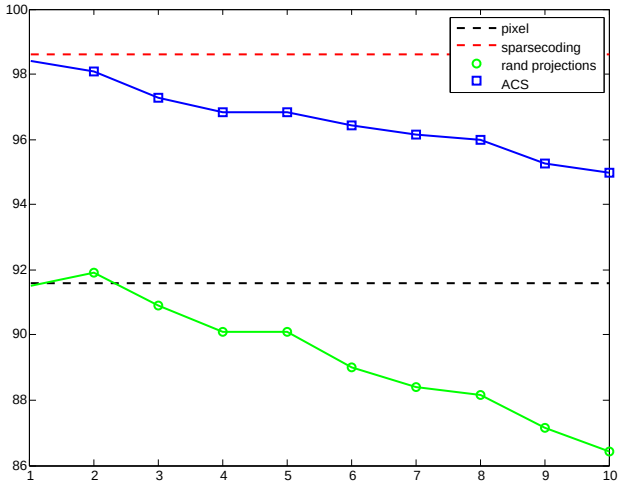


Figure A.5: Classification performance as a function of compression.

Once the model was trained, we loaded the test set and inferred sparse codes based on the previously learnt dictionary. We trained a linear SVM on the inferred coefficients of the train set and tested our model by applying the learnt SVM on the inferred codes of the test set. The results of our experiments are show in A.5. For reference points, we show the accuracy of a linear SVM on pixels (black dots) and sparse coding followed by a linear SVM on the original pixel input (red dots). In both cases the values were calculated only on the original input size of 28x28, i.e. no compression was done. The green curve shows the performance of a linear SVM on the random projected vectors. Note that ACS maintains a

We also explored various sizes of over completeness of basis and found that the best results for this varied from 4 times over complete to about 10 times over complete. While there was no obvious correlation our experiments seemed to suggest that for larger compressions a more over complete basis is required.

A.5 Discussion

The problem of extracting powerful features for classification has been engaging the fields of machine learning and computer vision for the past few decades. Although sparse dictionary learning has been shown to extract features that enable efficient classification [195, 17, 37, 157], high computational costs limit dictionary learning to rather small image patches, thus extracting quite local features. Here we demonstrate the possibility of using the principle of compressed sampling for dimensionality reduction. The theory of adaptive compressed sampling (ACS) guarantees that this form of compression allows for the extraction of non-local features at reduced computational costs.

Our experimental results suggest that ACS produces classification features that are interesting in various regards. First, they enable high classification accuracies and performance which degrades slowly with compression factor as long as the mathematical conditions of compressed sampling are still fulfilled. Second, the non-local ACS features capture the overall structure of the digits. Therefore, the set of class-specific features can be used to characterize shape properties of any given class, adding an interesting analytic or explanatory component to the classification.

In the future, we would like to test the presented method on the classification of more complex data, such as natural images, the CALTECH101 data, or in brain-machine interfaces using multi-electrode, ECoG,

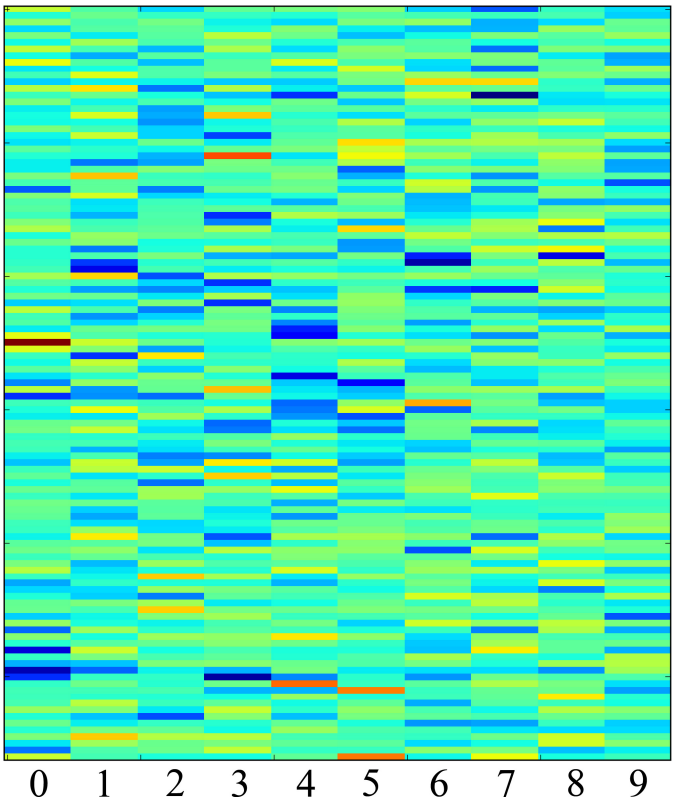


Figure A.6: **Non-local class-specific features emerge in a compressed space**: The figure above represents a coefficient $\times$ digit class matrix computed as follows. Given a fixed digit class and a neuron, we Z-score the set of coefficients inferred over compressed input images from the specified class. We then choose the absolute value of the biggest Z-scored coefficient in a given class as the entry of the above matrix. The colors (red and blue) represent values for coordinates which were very salient for the given class, . Some of the more salient structures picked out by a given neuron for a class are shown in Figure A.2. At the right, we have done a similar analysis on a local, convolutional style network .

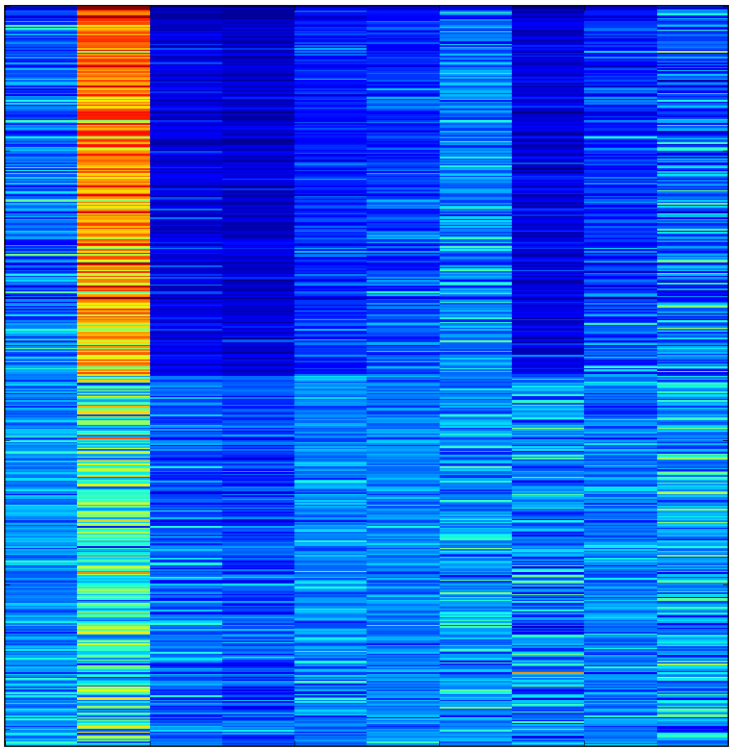


Figure A.7: **Local non-class-specific features**: To show class specificity we ran experiments on the MNIST dataset where we patched the original MNIST input image into 16 8x8 patches. The patches were then put through the ACS pipeline and a sparse model was learnt in the compressed domain. The class specificity plot was then computed similarly as in the case of the non-convolution (whole image). We note an apparent lack of structure in the z score, implying a more local structure.

or EEG data. Potentially, the extracted holistic classification features yield not only high classification performance but will also give new insights into the high-dimensional structure.

A.6 Acknowledgements

The work in this chapter was performed in collaboration with Aditya Joshi, Nicolas Mueler with advise from Chris Hillar, Fritz Sommer and Bruno Olshausen. This work was presented at the Neural Information Processing Systems workshop for the High-dimensional Statistical Inference in the Brain

Appendix B

Predicting V1 neural responses to natural movies using the shift-invariant bispectrum

Evidence from electrophysiology [154, 49] suggests that the visual system is highly sensitive to higher-order stimulus statistics. However, most models for the stimulus response of V1 neurons are limited to first- and second-order statistics, i.e. features defined on raw pixels or the power spectrum of the stimulus [41]. We explore the image bispectrum as a way to capture higher order features in the stimulus. We show that the performance of spiking response models can be improved by including these higher order features compared to only first and second order features. The bispectrum, which consists of the products of pairs of complex Fourier coefficients, has been used by researches in machine learning and image coding to produce invariant representations and characterize higher order image features such as curvature [104]. The elements of the bispectrum are translation-invariant like the power spectrum, yet retain relative phase information. This allows the bispectrum to capture features such as sharp edges, corners and T-junctions, which may underlie response properties of V1 cells.We test this hypothesis by fitting models to 128 cells recorded from cat and primate primary visual cortex. Three different models were fit to each cell: 1) raw pixels, 2) power spectrum and 3) the bispectrum of the stimulus movies. For 27/128 cells, the bispectrum model outperforms the pixel model and the power spectrum model. Thus, while the majority of cells can be better described as either simple cells with the pixel model or complex cells with the power spectrum model, a significant fraction (21%) of cells have more complex receptive fields and can be better modeled in terms of bispectrum features.

The bispectrum has been applied to translation and rotation invariant object reconstruction [164]. Figure B.1 shows a schematic that depicts how the bispectrum can be used to represent two objects which is rotation and translation invariant.

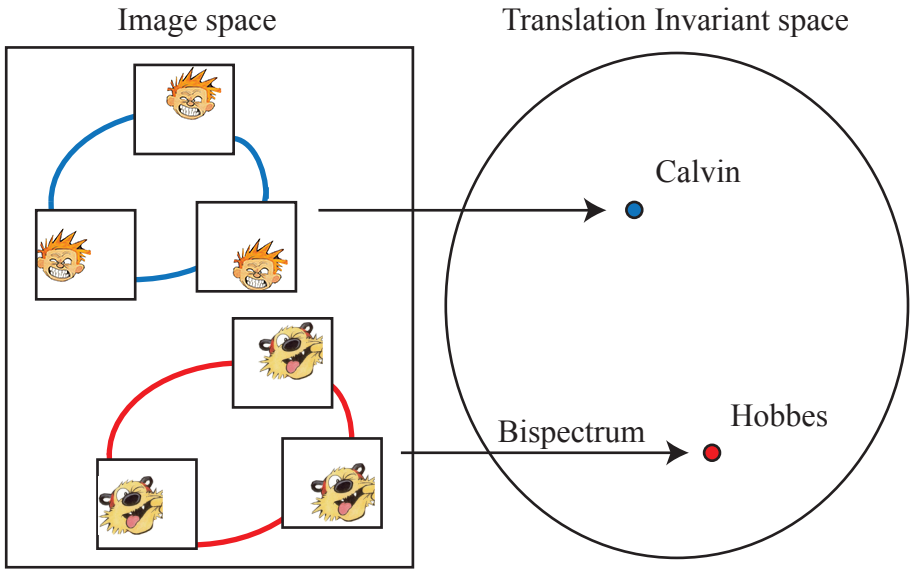


Figure B.1: A schematic describing translational and rotation invariance representation using the bispectrum for two different objects

A unique propert of the bispectrum is that when we average noisy signals in the bispectral space, we can recover the denoised version after inverting the bispectrum. This is shown in the Figure B.2. This is possible because the bispectrum retains the phase information.

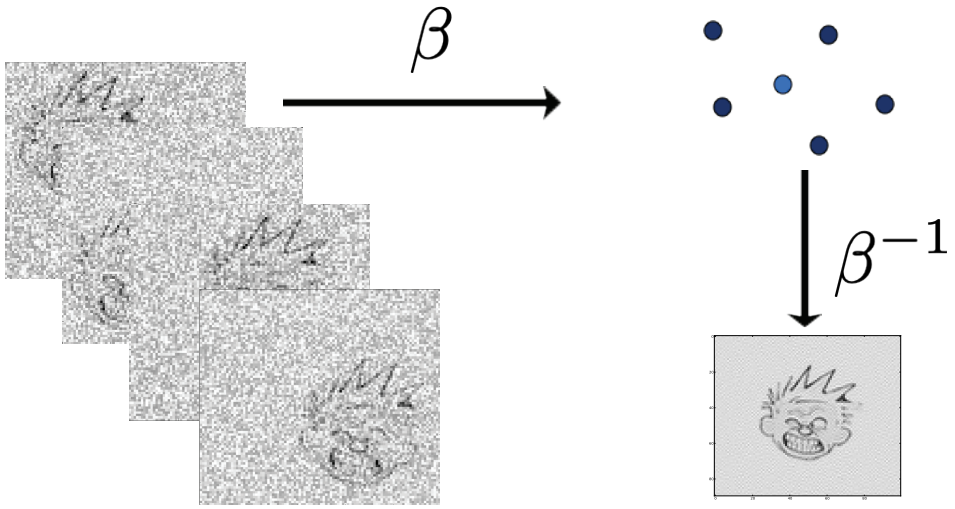


Figure B.2: Schematic explaining the idea of bispectrum for denoising. Here, we see that averaging in the bispectral space can lead to a cleaner version of the image because the bisectral coefficients retain phase information which is essential for signal recovery

B.1 Experimental procedures

Extracellular recordings in response to natural movie and image stimuli were made using silicon polytrodes and tetrodes in anesthetized cats and tungsten electrodes in behaving macaques (obtained from the CRCNS database). Spikes were binned at 30Hz to compute firing rates, either from single trials (polytrode data) or repeated presentation (tetrode and single electrodes).

The visual stimulus was cropped around the classical receptive field of the cells and down-sampled to 24x24 pixels. From this, we compute a 100-dimensional stimulus vector for each of the three different models: For the pixel model, we performed PCA to reduce the dimensionality to 100, for the Fourier power model, we performed PCA on the absolute value of the two-dimensional Fourier transform. Since the dimensionality of the bispectrum is the square of the size of the signal, an iterative approach was used to perform dimensionality reduction: For each pair of frequencies a 10-dimensional feature vector was computed using the largest principal components, then the dimensionality over all these feature vectors was reduced to 100 by performing PCA again. For the regression, each dimension was standardized to zero mean and unit variance. Gaussian ridge regression with cross validation was then used to estimate a linear filter in the feature space to optimally predict the firing rate response.

B.2 Model fitting

The 1D Fourier transform is given by,

$$\hat{f}(k) = \sum_{x=0}^{n-1} e^{-i2\pi xk/n} f(x) \tag{B.1}$$

The power spectrum of the Fourier transform is defined as the following

$$q(k) = \hat{f}(k)\overline{f(k)} = \|f(k)\|^2 \tag{B.2}$$

where the bar is the complex conjugate. The power spectrum is the Fourier transform of the autocorrelation function:

$$corr(x) = \sum_{y=0}^{n-1} f(y+x)f*(y) \tag{B.3}$$

The Bispectrum is defined as

$$b(k_1, k_2) = \mathcal{F}corr(x1, x2) \tag{B.4}$$

where we have the triplet correlation defined as

$$corr(x1, x2) = \sum_{y=0}^{n-1} f*(y-x_1)f*(y-x_2)f(y) \tag{B.5}$$

which gives us

$$\beta(k_1, k_2) = \hat{f}(k_1)\hat{f}(k_2)\overline{\hat{f}(k_1 + k_2)} \tag{B.6}$$

In amplitude and phase notation the fourier transform can be denoted as

$$\hat{f}(k) = a(k)e^{-j\alpha(k)} \tag{B.7}$$

and the power spectrum as

$$q(k) = a^2(k) \tag{B.8}$$

and the bispectrum can be defined as

$$\beta(k_1, k_2) = a(k_1)a(k_2)a(k_1 + k_2)e^{j(a(k_1)+a(k_2)-a(k_1+k_2))} \tag{B.9}$$

A shift in the signal by z gives

$$\hat{f}^z(k) = e^{-2\pi jzk/n}\hat{f}(k) \tag{B.10}$$

Substituting the above we get

$$\beta*(k_1, k_2) = e^{-2\pi jzk_1/n}\hat{f}(k_1)e^{-2\pi jzk_2/n}\hat{f}(k_2)e^{-2\pi jz(k_1+k_2)/n}\overline{\hat{f}(k_1 + k_2)} \tag{B.11}$$

Thus, we can see that the bispectrum gives us a local and global shift invariant representation while retaining the phase information. This is a key property.

B.3 Model comparison

Performance of first-order, second-order and bispectrum models for 128 cells recorded from primary visual cortex as seen in Figure B.3. In the top left, we see the log-likelihood of the three models separately for three groups of cells: 1) Those for which the bispectrum outperforms 1st and 2nd order models 2) those for which the bispectrum performs worse (simple or complex cells) and 3) all cells. In the top right of Figure B.3 shows the same comparison using correlation coefficients rather than log-likelihood. In c) we show examples of the response of neurons and models to one of the stimulus images. In the bottom of Figure B.3 the neural response was obtained by scanning the image over the receptive field of the cell following a Hilbert curve path. Model responses show that the bispectrum can learn features that are more selective than either the pixel or Fourier power model, and in good agreement with the neural response.

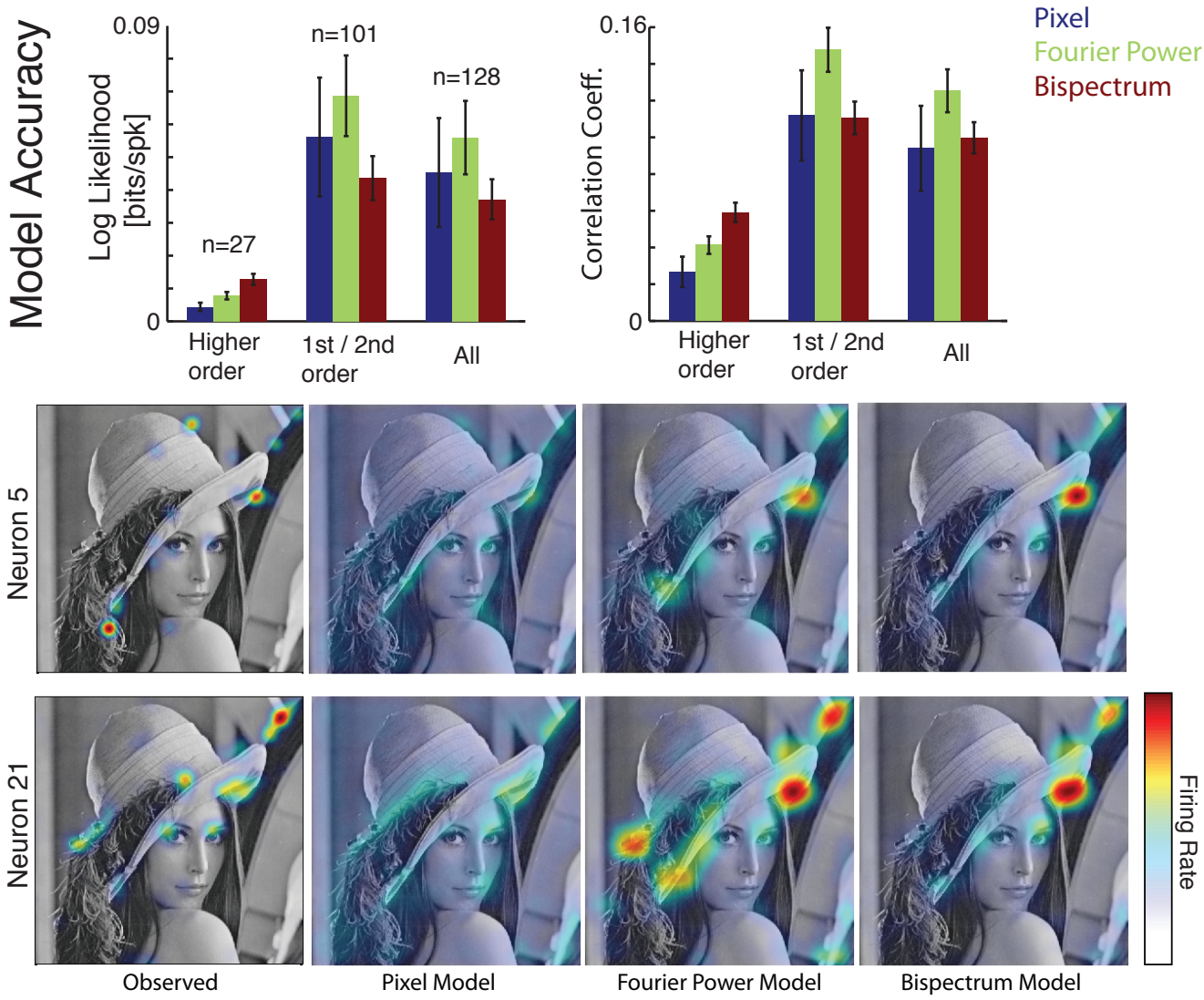


Figure B.3: Model comparison of 128 V1 neurons responding to natural scene movies. There are three groups in each of the plot based on the winning feature space. (Top Left) Log likelihood estimates. (Top Right) Correlation coefficients. (bottom rows) Convolving the receptive field with a test image with two different neurons.

B.4 Do random triplets work as well?

We wanted to explore if random triplets of pixels and FFT work just as well as the bispectral coefficients. Figure B.4 shows the results of this expeirment. We that the bispectrum outperforms other triplets of features.

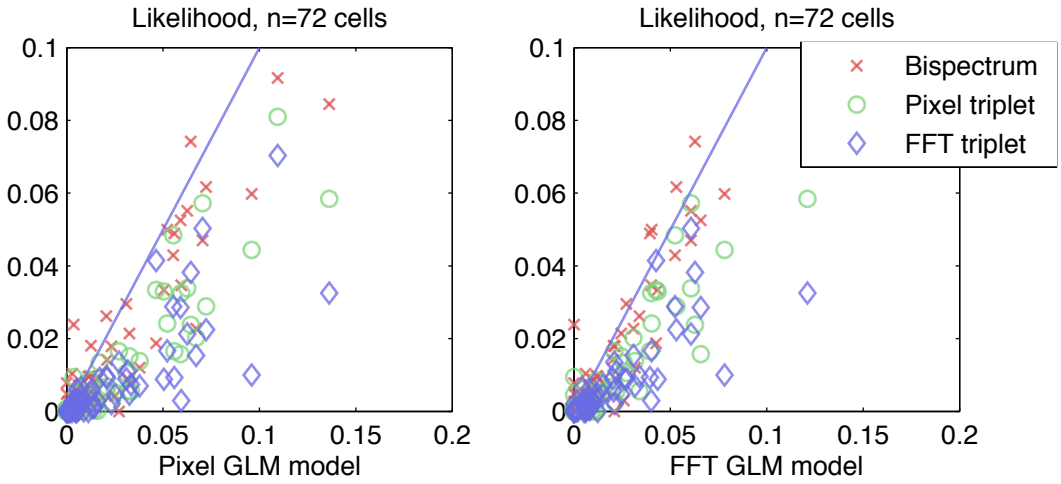


Figure B.4: Exponential GLM. The bispectrum outperforms features computed form random triplet correlations in pixel or FFT space.

B.5 Discussion

A subset of cells are better characterized in terms of bispectrum as opposed to simple and complex cell models. Future recordings might employ stimuli specifically designed to probe differences in response to relative and global phase components. We found that the Fourier basis is not appropriate for natural scenes. Generalization to other basis with this model could prove more enlightening

B.6 Acknowledgements

The work in this chapter was performed in collaboration with Ian Stevenson, Urs Koster, and Chris Hillar with advise from Bruno Olshausen. This work was presented at the annual Computational and Systems Neuroscience Conference at Salt Lake City, Utah, March 2013.

Bibliography

[1] *A Compilation of Robots Falling Down at the DARPA Robotics Challenge*. Youtube. 2015. URL: https://www.youtube.com/watch?v=g0TaYhjpOfo.

[2] Salvatore Aglioti, Joseph FX DeSouza, and Melvyn A Goodale. "Size-contrast illusions deceive the eye but not the hand". In: *Current biology* 5.6 (1995), pp. 679–685.

[3] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. "Learning to see by moving". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 37–45.

[4] M. Aharon, M. Elad, and A.M. Bruckstein. "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them". In: *Linear Algebra and its Applications* 416.1 (2006), pp. 48–67. ISSN: 0024-3795.

[5] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. "Active vision". In: *International journal of computer vision* 1.4 (1988), pp. 333–356.

[6] David F Anderson and Thomas G Kurtz. "Continuous time Markov chain models for chemical reaction networks". In: *Design and analysis of biomolecular circuits*. Springer, 2011, pp. 3–42.

[7] Joseph J Atick and A Norman Redlich. "What does the retina know about natural scenes?" In: *Neural computation* 4.2 (1992), pp. 196–210.

[8] K. Balakrishnan. *Exponential Distribution: Theory, Methods and Applications*. CRC Press, 1996, p. 664. ISBN: 2884491929.

[9] R. Baraniuk et al. "A simple proof of the restricted isometry property for random matrices". In: *Constructive Approximation* 28.3 (2008), pp. 253–263.

[10] Horace B Barlow et al. "Possible principles underlying the transformation of sensory messages". In: *Sensory communication* 1 (1961), pp. 217–234.

[11] A. Bell and T. Sejnowski. "Learning the higher-order structure of a natural sound". In: *Network: Computation in Neural Systems* 7.2 (1996), pp. 261–266.

[12] Anthony J Bell and Terrence J Sejnowski. "The "independent components" of natural scenes are edge filters". In: *Vision research* 37.23 (1997), pp. 3327–3338.

[13] A Beskos et al. "Hybrid monte carlo on hilbert spaces". In: *Stochastic Processes and their Applications* (2011).

[14] M J Betancourt et al. "The Geometric Foundations of Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1410.5110* (2014).

[15] Volker Blanz, Thomas Vetter, et al. "A morphable model for the synthesis of 3D faces." In: *Siggraph*. Vol. 99. 1999. 1999, pp. 187–194.

[16] Hermann Borotschnig et al. "Active Object Recognition in Parametric Eigenspace." In: *BMVC*. Citeseer. 1998, pp. 1–10.

[17] Y-L Boureau et al. "Learning mid-level features for recognition". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 2559–2566.

[18] Franz Clemens Brentano. *Psychology from the empirical standpoint*. Vol. 1. Duncker & Humblot, 1874.

[19] CF Cadieu and K Koepsell. "Phase coupling estimation from multivariate phase statistics". In: *Neural computation* (2010). URL: http://www.mitpressjournals.org/doi/abs/10.1162/NECO%5C_a%5C_00048.

[20] Charles F Cadieu and Bruno A Olshausen. "Learning intermediate-level representations of form and motion from natural movies". In: *Neural computation* 24.4 (2012), pp. 827–866.

[21] Roberto Calandra et al. "More Than a Feeling: Learning to Grasp and Regrasp using Vision and Touch". In: *IEEE Robotics and Automation Letters (RA-L)* 3.4 (2018), pp. 3300–3307. DOI: 10.1109/LRA.2018.2852779.

[22] Roberto Calandra et al. "The Feeling of Success: Does Touch Sensing Help Predict Grasp Outcomes?" In: *Conference on Robot Learning (CORL)* (2017).

[23] Cédric M Campos and J M Sanz-Serna. "Extra Chance Generalized Hybrid Monte Carlo". In: *Journal of Computational Physics* 281 (2015), pp. 365–374.

[24] CM Campos and JM Sanz-Serna. "Extra Chance Hybrid Monte Carlo". In: *arXiv preprint arXiv:1407.8107* (2014). URL: http://arxiv.org/abs/1407.8107.

[25] E.J. Candes and T. Tao. "Decoding by linear programming". In: *Information Theory, IEEE Transactions on* 51.12 (2005), pp. 4203–4215. ISSN: 0018-9448.

[26] John Canny. "A computational approach to edge detection". In: *Readings in computer vision*. Elsevier, 1987, pp. 184–203.

[27] Nicole L. Carlson, Vivienne L. Ming, and Michael Robert DeWeese. "Sparse Codes for Speech Predict Spectrotemporal Receptive Fields in the Inferior Colliculus". In: *PLoS Comput Biol* 8.7 (2012), e1002594. DOI: 10.1371/journal.pcbi.1002594.

[28] Stefano Caselli, Corrado Magnanini, and Francesco Zanichelli. "Haptic object recognition with a dextrous hand based on volumetric shape representations". In: *Multisensor Fusion and Integration for Intelligent Systems, 1994. IEEE International Conference on MFI'94.* IEEE. 1994, pp. 280–287.

[29] Angel X Chang et al. "Shapenet: An information-rich 3d model repository". In: *arXiv preprint arXiv:1512.03012* (2015).

[30] Wei-Lun Chao et al. "Exponential Integration for Hamiltonian Monte Carlo". In: *International Conference on Machine Learning* (2015).

[31] Yevgen Chebotar et al. "Bigs: Biotac grasp stability dataset". In: *ICRA 2016 Workshop on Grasping and Manipulation Datasets*. 2016.

[32] Tianqi Chen, Emily B Fox, and Carlos Guestrin. "Stochastic gradient hamiltonian monte carlo". In: *arXiv preprint arXiv:1402.4102* (2014).

[33] Vivian Chu et al. "Using robotic exploratory procedures to learn the meaning of haptic adjectives". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 3048–3055.

[34] Leon O Chua and Sung Mo Kang. "Memristive devices and systems". In: *Proceedings of the IEEE* 64.2 (1976), pp. 209–223.

[35] Erhan Cinlar. *Introduction to stochastic processes*. Courier Corporation, 2013.

[36] David L Clark and George W Uetz. "Video image recognition by the jumping spider, Maevia inclemens (Araneae: Salticidae)". In: *Animal behaviour* 40.5 (1990), pp. 884–890.

[37] Adam Coates and Andrew Ng. "The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization". In: *Proc. of the 28th International Conference on Machine Learning (ICML)*. Ed. by Lise Getoor and Tobias Scheffer. New York, NY, 2011, pp. 921–928.

[38] W.K. Coulter et al. "Adaptive compressed sensing: A new class of self-organizing coding models for neuroscience". In: *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE. 2010, pp. 5494–5497.

[39] Benjamin J Culpepper, Jascha Sohl-Dickstein, and Bruno A Olshausen. "Building a better probabilistic model of images by factorization". In: *International Conference on Computer Vision* (2011).

[40] Ravinder S Dahiya et al. "Tactile sensing—from humans to humanoids". In: *IEEE transactions on robotics* 26.1 (2010), pp. 1–20.

[41] Stephen V David and Jack L Gallant. "Predicting neuronal responses during natural vision". In: *Network: Computation in Neural Systems* 16.2-3 (2005), pp. 239–260.

[42] Raphael Deimel and Oliver Brock. "A novel type of compliant and underactuated robotic hand for dexterous grasping". In: *The International Journal of Robotics Research* 35.1-3 (2016), pp. 161–185.

[43] Raphael Deimel et al. "Automated Co-Design of Soft Hand Morphology and Control Strategy for Grasping". In: ().

[44] Aaron M Dollar and Robert D Howe. "Simple, robust autonomous grasping in unstructured environments". In: *Robotics and Automation, 2007 IEEE International Conference on*. IEEE. 2007, pp. 4693–4700.

[45] D.L. Donoho. "Compressed sensing". In: *Information Theory, IEEE Transactions on* 52.4 (2006), pp. 1289–1306. ISSN: 0018-9448.

[46] S Duane et al. "Hybrid monte carlo". In: *Physics letters B* (1987).

[47] BBC Earth. *Spider With Three Super Powers — The Hunt — BBC Earth*. July 2017. URL: `https://www.youtube.com/watch?v=UDtlvZGmHYk`.

[48] Frederik Ebert et al. "Self-Supervised Visual Planning with Temporal Skip Connections". In: *CORL* abs/1710.05268 (2017). arXiv: `1710.05268`. URL: `http://arxiv.org/abs/1710.05268`.

[49] Gidon Felsen et al. "Cortical sensitivity to visual features in natural scenes". In: *PLoS biology* 3.10 (2005), e342.

[50] Kristen A Fichthorn and W Hh Weinberg. "Theoretical foundations of dynamical Monte Carlo simulations". In: *The Journal of chemical physics* 95.2 (1991), pp. 1090–1096.

[51] David J Field. "Relations between the statistics of natural images and the response properties of cortical cells". In: *Josa a* 4.12 (1987), pp. 2379–2394.

[52] Chelsea Finn, Ian Goodfellow, and Sergey Levine. "Unsupervised learning for physical interaction through video prediction". In: *Advances in neural information processing systems*. 2016, pp. 64–72.

[53] Chelsea Finn and Sergey Levine. "Deep Visual Foresight for Planning Robot Motion". In: *International Conference on Robotics and Automation (ICRA)*. 2017.

[54] Jeremy A Fishel and Gerald E Loeb. "Bayesian exploration for intelligent identification of textures". In: *Frontiers in neurorobotics* 6 (2012), p. 4.

[55] Jeremy A Fishel and Gerald E Loeb. "Sensing tactile microvibrations with the BioTac—Comparison with human sensitivity". In: *IEEE RAS & EMBS International Biomedical Robotics and Biomechatronics (BioRob)*. 2012, pp. 1122–1127.

[56] Jeremy Fishel, Gary Lin, and GE Loeb. "Syntouch LLC biotac product manual, v. 16". In: *Tech. Rep.* (2013).

[57] Surya Ganguli and Haim Sompolinsky. "Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis". In: *Annual Review of Neuroscience* 35 (2012), pp. 485–508.

[58] Yang Gao et al. "Deep learning for tactile understanding from visual and haptic data". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 536–543.

[59] Pierre Garrigues and Bruno A Olshausen. "Group sparse coding with a laplacian scale mixture prior". In: *Advances in neural information processing systems*. 2010, pp. 676–684.

[60] James J Gibson. "Observations on active touch." In: *Psychological review* 69.6 (1962), p. 477.

[61] James J Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.

[62] James J Gibson. "The theory of affordances". In: *Hilldale, USA* 1 (1977), p. 2.

[63] Daniel T Gillespie. "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions". In: *Journal of computational physics* 22.4 (1976), pp. 403–434.

[64] Mark Girolami and Ben Calderhead. "Riemann manifold Langevin and Hamiltonian Monte Carlo methods". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2 (Mar. 2011), pp. 123–214. ISSN: 13697412. DOI: 10.1111/j.1467-9868.2010.00765.x.

[65] Joshua I Glaser et al. "Machine learning for neural decoding". In: *arXiv preprint arXiv:1708.00909* (2017).

[66] S. Gleichman and Y.C. Eldar. "Blind compressed sensing". In: *Information Theory, IEEE Transactions on* 57.10 (2011), pp. 6958–6975.

[67] Kenneth Y Goldberg and Ruzena Bajcsy. "Active touch and robot perception". In: *Cognition and Brain Theory* 7.2 (1984), pp. 199–214.

[68] Corey Goldfeder et al. "The columbia grasp database". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 1710–1716.

[69] Melvyn A Goodale and A David Milner. "Separate visual pathways for perception and action". In: *Trends in neurosciences* 15.1 (1992), pp. 20–25.

[70] Melvyn A Goodale et al. "A neurological dissociation between perceiving objects and grasping them". In: *Nature* 349.6305 (1991), p. 154.

[71] Nicolas Gorges et al. "Haptic object recognition using passive joints and haptic key features". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 2349–2355.

[72] U Grenander and MI Miller. "Representations of knowledge in complex systems". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1994).

[73] Marcus Gualtieri et al. "High precision grasp pose detection in dense clutter". In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE. 2016, pp. 598–605.

[74] Abhishek Gupta et al. "Learning dexterous manipulation for a soft robotic hand from human demonstrations". In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE. 2016, pp. 3786–3793.

[75] Tuomas Haarnoja et al. "Reinforcement learning with deep energy-based policies". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1352–1361.

[76] E Hairer, C Lubich, and G Wanner. "Geometric numerical integration illustrated by the Stormer-Verlet method". In: *Acta Numerica* (2003). URL: http://journals.cambridge.org/production/action/cjoGetFulltext?fulltextid=165634.

[77] Liberty S Hamilton et al. "Optogenetic activation of an inhibitory network enhances feedforward functional connectivity in auditory cortex". In: *Neuron* 80.4 (2013), pp. 1066–1076.

[78] Mallory L Hammock et al. "25th anniversary article: the evolution of electronic skin (e-skin): a brief history, design considerations, and recent progress". In: *Advanced materials* 25.42 (2013), pp. 5997–6038.

[79] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[80] W Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* (Jan. 1970).

[81] Alexander Herzog et al. "Learning of grasp selection based on shape-templates". In: *Autonomous Robots* 36.1-2 (2014), pp. 51–65.

[82] C. Hillar and F. T. Sommer. "When can dictionary learning uniquely recover sparse data from subsamples?" In: *ArXiv e-prints* (2013). arXiv: 1106.3616.

[83] MD Hoffman and A Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Arxiv preprint arXiv:1111.4246* (2011), pp. 1–30. arXiv: arXiv:1111.4246v1. URL: http://arxiv.org/abs/1111.4246.

[84] Herke van Hoof et al. "Stable reinforcement learning with autoencoders for tactile and visual data". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 3928–3934.

[85] AM Horowitz. "A generalized guided Monte Carlo algorithm". In: *Physics Letters B* (1991). URL: http://www.sciencedirect.com/science/article/pii/0370269391908125.

[86] David H Hubel and Torsten N Wiesel. "Receptive fields and functional architecture of monkey striate cortex". In: *The Journal of physiology* 195.1 (1968), pp. 215–243.

[87] David H Hubel and Torsten N Wiesel. "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of physiology* 148.3 (1959), pp. 574–591.

[88] James M Hughes, Daniel J Graham, and Daniel N Rockmore. "Quantification of artistic style through sparse coding analysis in the drawings of Pieter Bruegel the Elder". In: *Proc. of the National Academy of Sciences* 107.4 (2010), pp. 1279–1283.

[89] Akihisa Ichiki and Masayuki Ohzeki. "Violation of detailed balance accelerates relaxation". In: *Physical Review E* 88.2 (Aug. 2013), p. 020101. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.88.020101.

[90] Guy Isely, Christopher Hillar, and Fritz Sommer. "Deciphering subsampled data: adaptive compressive sampling as a principle of brain communication". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. Lafferty et al. 2010, pp. 910–918.

[91] JA Izaguirre and SS Hampton. "Shadow hybrid Monte Carlo: an efficient propagator in phase space of macromolecules". In: *Journal of Computational Physics* (2004).

[92] Gregory Izatt et al. "Tracking objects with point clouds from vision and touch". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4000–4007.

[93] Dinesh Jayaraman and Kristen Grauman. "End-to-end policy learning for active visual categorization". In: *IEEE transactions on pattern analysis and machine intelligence* (2018).

[94] Dinesh Jayaraman and Kristen Grauman. "Learning image representations tied to ego-motion". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1413–1421.

[95] *Johansson Coding*. Youtube. 2011. URL: `https://www.youtube.com/watch?v=0LfJ3M3Kn80`.

[96] Roland S Johansson and J Randall Flanagan. "Coding and use of tactile signals from the fingertips in object manipulation tasks". In: *Nature Reviews Neuroscience* 10.5 (2009).

[97] Micah K Johnson et al. "Microgeometry capture using an elastomeric sensor". In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 4. ACM. 2011, p. 46.

[98] Michael I Jordan et al. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2 (1999), pp. 183–233.

[99] Daniel Kappler, Jeannette Bohg, and Stefan Schaal. "Leveraging big data for grasp planning". In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 4304–4311.

[100] S Karlin. *A first course in stochastic processes*. 1968, pp. 206–223.

[101] AD Kennedy and B Pendleton. "Acceptances and autocorrelations in hybrid Monte Carlo". In: *Nuclear Physics B-Proceedings Supplements* (1991).

[102] Daniel Kersten, Pascal Mamassian, and Alan Yuille. "Object perception as Bayesian inference". In: *Annu. Rev. Psychol.* 55 (2004), pp. 271–304.

[103] David C Knill and Whitman Richards. *Perception as Bayesian inference*. Cambridge University Press, 1996.

[104] Gerhard Krieger, Christoph Zetzsche, and Erhardt Barth. "Higher-order statistics of natural images and their exploitation by operators selective to intrinsic dimensionality". In: *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics*. IEEE. 1997, pp. 147–151.

[105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[106] Vikash Kumar. *SHAP arm implementation in Mujoco*. 2016. URL: `http://www.mujoco.org/forum/index.php?resources/modular-prosthetic-limb-shap-test-suites.19/`.

[107] Vikash Kumar, Zhe Xu, and Emanuel Todorov. "Fast, strong and compliant pneumatic actuation for dexterous tendon-driven hands". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 1512–1519.

[108] Vikash Kumar et al. "Learning Dexterous Manipulation Policies from Experience and Imitation". In: *arXiv preprint arXiv:1611.05095* (2016).

[109] Kai Labusch, Erhardt Barth, and Thomas Martinetz. "Simple method for high-performance digit recognition based on sparse coding". In: *Neural Networks, IEEE Transactions on* 19.11 (2008), pp. 1985–1989.

[110] Thomas Lampe and Martin Riedmiller. "Acquiring visual servoing reaching and grasping skills using neural reinforcement learning". In: *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE. 2013, pp. 1–8.

[111] MF Land. "The morphology and optics of spider eyes". In: *Neurobiology of arachnids*. Springer, 1985, pp. 53–78.

[112] Quoc V Le et al. "Building high-level features using large scale unsupervised learning". In: *arXiv preprint arXiv:1112.6209* (2011).

[113] Y. LeCun and C. Cortes. *The MNIST database of handwritten digits*. 1998.

[114] Susan J Lederman and Roberta L Klatzky. "Extracting object properties through haptic exploration". In: *Acta psychologica* 84.1 (1993), pp. 29–40.

[115] Susan J Lederman and Roberta L Klatzky. "Hand movements: A window into haptic object recognition". In: *Cognitive psychology* 19.3 (1987), pp. 342–368.

[116] Alex X Lee et al. "Stochastic Adversarial Video Prediction". In: *arXiv preprint arXiv:1804.01523* (2018).

[117] Nathan F Lepora, Kirsty Aquilina, and Luke Cramphorn. "Exploratory Tactile Servoing With Active Touch." In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 1156–1163.

[118] Sergey Levine et al. "End-to-end Training of Deep Visuomotor Policies". In: *J. Mach. Learn. Res.* 17.1 (Jan. 2016), pp. 1334–1373. ISSN: 1532-4435.

[119] Sergey Levine et al. "End-to-end training of deep visuomotor policies". In: *arXiv preprint arXiv:1504.00702* (2015).

[120] Michael S Lewicki et al. "Scene analysis in the natural environment". In: *Frontiers in psychology* 5 (2014), p. 199.

[121] Qiang Li et al. "A control framework for tactile servoing". In: *Robotics:Science and Systems (RSS)* (2013).

[122] Rui Li et al. "Localization and manipulation of small parts using gelsight tactile sensing". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2014, pp. 3988–3993.

[123] Colin M Light, Paul H Chappell, and Peter J Kyberd. "Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: normative data, reliability, and validity". In: *Archives of physical medicine and rehabilitation* 83.6 (2002), pp. 776–783.

[124] Timothy P Lillicrap et al. "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971* (2015).

[125] DJC MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[126] Jeffrey Mahler et al. "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics". In: *arXiv preprint arXiv:1703.09312* (2017).

[127] Vikash K Mansinghka, Eric M Jonas, and Joshua B Tenenbaum. "Stochastic digital circuits for probabilistic inference". In: *Massachussets Institute of Technology, Technical Report MITCSAIL-TR* 2069 (2008).

[128] Enzo Marinari and Giorgio Parisi. "Simulated tempering: a new Monte Carlo scheme". In: *EPL (Europhysics Letters)* 19.6 (1992), p. 451.

[129] David Marr and Ellen Hildreth. "Theory of edge detection". In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 207.1167 (1980), pp. 187–217.

[130] Blaine Matulevich, Gerald E Loeb, and Jeremy A Fishel. "Utility of contact detection reflexes in prosthetic hand control". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013, pp. 4741–4746.

[131] Ferenc Matyas et al. "Motor control by sensory cortex". In: *Science* 330.6008 (2010), pp. 1240–1243.

[132] Edward Meeds, Robert Leenders, and Max Welling. "Hamiltonian ABC". In: *arXiv preprint arXiv:1503.01916* (2015).

[133] Philipp Metzner, Christof Schütte, and Eric Vanden-Eijnden. "Transition path theory for Markov jump processes". In: *Multiscale Modeling & Simulation* 7.3 (2009), pp. 1192–1219.

[134] Volodymyr Mnih et al. "Asynchronous methods for deep reinforcement learning". In: *ICML*. 2016.

[135] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* (2015).

[136] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.

[137] Richard M Murray et al. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[138] Radford M Neal. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* (Jan. 2010).

[139] RM Neal. "An improved acceptance procedure for the hybrid Monte Carlo algorithm". In: *Journal of Computational Physics* (1994).

[140] RM Neal. "Probabilistic inference using Markov chain Monte Carlo methods". In: *Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto* (1993).

[141] Cristopher M Niell and Michael P Stryker. "Modulation of visual responses by behavioral state in mouse visual cortex". In: *Neuron* 65.4 (2010), pp. 472–479.

[142] Lael U Odhner et al. "A compliant, underactuated hand for robust manipulation". In: *The International Journal of Robotics Research* 33.5 (2014), pp. 736–752.

[143] B.A. Olshausen and D.J. Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583 (1996), pp. 607–609.

[144] et al. OpenAI: Marcin Andrychowicz. "Learning Dexterous In-Hand Manipulation". In: *arXiv:1808.00177* (2018).

[145] J Kevin O'Regan and Alva Noë. "A sensorimotor account of vision and visual consciousness". In: *Behavioral and brain sciences* 24.5 (2001), pp. 939–973.

[146] Simon Osindero, Max Welling, and Geoffrey E Hinton. "Topographic product models applied to natural scene statistics". In: *Neural Computation* 18.2 (2006), pp. 381–414.

[147] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. "Intrinsic motivation systems for autonomous mental development". In: *IEEE transactions on evolutionary computation* 11.2 (2007), pp. 265–286.

[148] Pierre-Yves Oudeyer and Frederic Kaplan. "What is intrinsic motivation? A typology of computational approaches". In: *Frontiers in neurorobotics* 1 (2009), p. 6.

[149] Andrew Owens et al. "Ambient sound provides supervision for visual learning". In: *European Conference on Computer Vision*. Springer. 2016, pp. 801–816.

[150] Deepak Pathak et al. "Curiosity-driven exploration by self-supervised prediction". In: *International Conference on Machine Learning (ICML)*. Vol. 2017. 2017.

[151] Dejan Pecevski, Lars Buesing, and Wolfgang Maass. "Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons". In: *PLoS computational biology* 7.12 (2011), e1002294.

[152] Lerrel Pinto and Abhinav Gupta. "Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours". In: *ICRA* (2016).

[153] Biermann PJ. "The Cosmesis: A Social and Functional Interface". In: *Johns Hopkins APL Tech Dig* 30.3 (2011), pp. 250–255.

[154] Keith P Purpura, Jonathan D Victor, and Ephraim Katz. "Striate cortex extracts higher-order spatial correlations from visual textures". In: *Proceedings of the National Academy of Sciences* 91.18 (1994), pp. 8482–8486.

[155] Vinayak Rao and Yee Whye Teh. "Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks". In: *arXiv preprint arXiv:1202.3760* (2012).

[156] A. Rehn and F. Sommer. "A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields". In: *Journal of Computational Neuroscience* 22.2 (2007), pp. 135–146.

[157] R. Rigamonti, M.A. Brown, and V. Lepetit. "Are sparse representations really relevant for image classification?" In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 1545–1552.

[158] Alberto Rodriguez, Matthew T Mason, and Steve Ferry. "From caging to grasping". In: *The International Journal of Robotics Research* 31.7 (2012), pp. 886–900.

[159] Joseph M Romano et al. "Human-inspired robotic grasp control with tactile sensing". In: *IEEE Transactions on Robotics* 27.6 (2011), pp. 1067–1079.

[160] Frank Rothling et al. "Platform portable anthropomorphic grasping with the bielefeld 20-dof shadow and 9-dof tum hand". In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE. 2007, pp. 2951–2956.

[161] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[162] D.L. Ruderman and W. Bialek. "Statistics of natural images: Scaling in the woods". In: *Physical Review Letters* 73.6 (1994), pp. 814–817.

[163] Daniela Rus and Michael T Tolley. "Design, fabrication and control of soft robots". In: *Nature* 521.7553 (2015), p. 467.

[164] Brian M Sadler and Georgios B Giannakis. "Shift-and rotation-invariant object reconstruction using the bispectrum". In: *JOSA A* 9.1 (1992), pp. 57–69.

[165] Jürgen Schmidhuber. "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)". In: *IEEE Transactions on Autonomous Mental Development* 2.3 (2010), pp. 230–247.

[166] Alexander Schneider et al. "Object identification with tactile sensors using bag-of-features". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE. 2009, pp. 243–248.

[167] Elad Schneidman et al. "Weak pairwise correlations imply strongly correlated network states in a neural population". In: *Nature* 440.7087 (2006), p. 1007.

[168] John Schulman et al. "Trust region policy optimization". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 2015, pp. 1889–1897.

[169] C Schütte and A Fischer. "A direct approach to conformational dynamics based on hybrid Monte Carlo". In: *Journal of Computational Physics* (1999).

[170] B Shahbaba et al. "Split hamiltonian monte carlo". In: *Statistics and Computing* (2011).

[171] S Murray Sherman and Rainer W Guillery. *Exploring the thalamus and its role in cortical function*. MIT press, 2006.

[172] Pavan Sikka, Hong Zhang, and Steve Sutphen. "Tactile servo: Control of touch-driven robot motion". In: *Experimental Robotics III*. Springer, 1994, pp. 219–233.

[173] Jivko Sinapov et al. "Vibrotactile recognition and categorization of surfaces by a humanoid robot". In: *IEEE Transactions on Robotics* 27.3 (2011), pp. 488–497.

[174] Adrian FM Smith and Gareth O Roberts. "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.1 (1993), pp. 3–23.

[175] E.C. Smith and M.S. Lewicki. "Efficient auditory coding". In: *Nature* 439.7079 (2006), pp. 978–982.

[176] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical Bayesian optimization of machine learning algorithms". In: *Advances in Neural Information Processing Systems*. 2012, pp. 2951–2959.

[177] Jascha Sohl-Dickstein. "Hamiltonian Monte Carlo with Reduced Momentum Flips". In: *Redwood Center Technical Report* arXiv 1205.1939 (May 2012). arXiv: 1205.1939. URL: http://arxiv.org/abs/1205.1939.

[178] Jascha Sohl-Dickstein, Peter B Battaglino, and Michael R DeWeese. "New method for parameter estimation in probabilistic models: minimum probability flow". In: *Physical review letters* 107.22 (2011), p. 220601.

[179] Jascha Sohl-Dickstein and Benjamin J. Culpepper. "Hamiltonian Annealed Importance Sampling for partition function estimation". In: *Redwood Center Technical Report* arXiv 1205.1925 (May 2012). arXiv: 1205.1925. URL: http://arxiv.org/abs/1205.1925.

[180] Jascha Sohl-Dickstein, Mayur Mudigonda, and Michael DeWeese. "Hamiltonian Monte Carlo without detailed balance". In: *Proceedings of the 31st International Conference on Machine Learning*. 2014, pp. 719–726.

[181] Daniel A. Spielman, Huan Wang, and John Wright. "Exact Recovery of Sparsely-Used Dictionaries". In: *Journal of Machine Learning Research - Proceedings Track* 23 (2012), pp. 37.1–37.18.

[182] Subramanian Sundaram et al. "Learning the signatures of the human grasp using a scalable tactile glove". In: *Nature* 569.7758 (2019), p. 698.

[183] Giovanni Sutanto et al. "Learning Latent Space Dynamics for Tactile Servoing". In: *arXiv preprint arXiv:1811.03704* (2018).

[184] Emanuel Todorov, Tom Erez, and Yuval Tassa. "MuJoCo: A physics engine for model-based control". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.

[185] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors". In: *IEEE transactions on pattern analysis and machine intelligence* 30.5 (2008), pp. 878–892.

[186] Filipe Veiga et al. "Tactile based Forward Modeling for Contact Location Control". In: *RSS Workshop on Tactile Sensing for Manipulation* (2017).

[187] Arthur F Voter. "Introduction to the kinetic Monte Carlo method". In: *Radiation Effects in Solids*. Springer, 2007, pp. 1–23.

[188] M.J. Wainwright. "Sharp thresholds for high-dimensional and noisy sparsity recovery using ell_1-constrained quadratic programming (Lasso)". In: *IEEE Trans. Information Theory* (2009), pp. 2183–2202.

[189] Vincent Wall, Gabriel Zöller, and Oliver Brock. "A method for sensorizing soft actuators and its application to the RBO hand 2". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE. 2017, pp. 4965–4970.

[190] Tianshi Wang and Jaijeet Roychowdhury. "Oscillator-based ising machine". In: *arXiv preprint arXiv:1709.08102* (2017).

[191] Z Wang, S Mohamed, and D Nando. "Adaptive Hamiltonian and Riemann Manifold Monte Carlo". In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (2013).

[192] Douglas Weber. *Hand Proprioception and touch Interfaces*. URL: `http://www.darpa.mil/program/hand-proprioception-and-touch-interfaces`.

[193] Elizabeth L Wilmer. "Markov Chains and Mixing Times David A . Levin Yuval Peres". In: (2009).

[194] Zhe Xu, Vikash Kumar, and Emanuel Todorov. "The UW Hand: A Low-cost, 20-DOF Tendon-driven Hand with Fast and Compliant Actuation". In: *The International Journal of Robotics Research* (2013).

[195] Jianchao Yang et al. "Linear spatial pyramid matching using sparse coding for image classification". In: *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on*. IEEE. 2009, pp. 1794–1801.

[196] Jin Yang and William S Hlavacek. "Rejection-free kinetic Monte Carlo simulation of multivalent biomolecular interactions". In: *arXiv preprint arXiv:0812.4619* (2008).

[197] Jin Yang et al. "Kinetic Monte Carlo method for rule-based modeling of biochemical networks". In: *Physical Review E* 78.3 (2008), p. 031910.

[198] Hanna Yousef, Mehdi Boukallel, and Kaspar Althoefer. "Tactile sensing for dexterous in-hand manipulation in robotics — A review". In: *Sensors and Actuators A: physical* 167.2 (2011), pp. 171–187.

[199] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. "GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force". In: *Sensors* (2017). DOI: 10.3390/s17122762.

[200] Wenzhen Yuan et al. "Shape-independent hardness estimation using deep learning and a GelSight tactile sensor". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 951–958.

[201] Richard Zhang, Phillip Isola, and Alexei A Efros. "Colorful image colorization". In: *European Conference on Computer Vision*. Springer. 2016, pp. 649–666.

[202] Y Zhang and C Sutton. "Quasi-Newton Markov chain Monte Carlo". In: (2011).

[203] David Zipser and Richard A Andersen. "A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons". In: *Nature* 331.6158 (1988), p. 679.

[204] Liang Zou et al. "Novel tactile sensor technology and smart tactile sensing systems: A review". In: *Sensors* 17.11 (2017), p. 2653.